

*Fig. 1*  
*Prior Art*

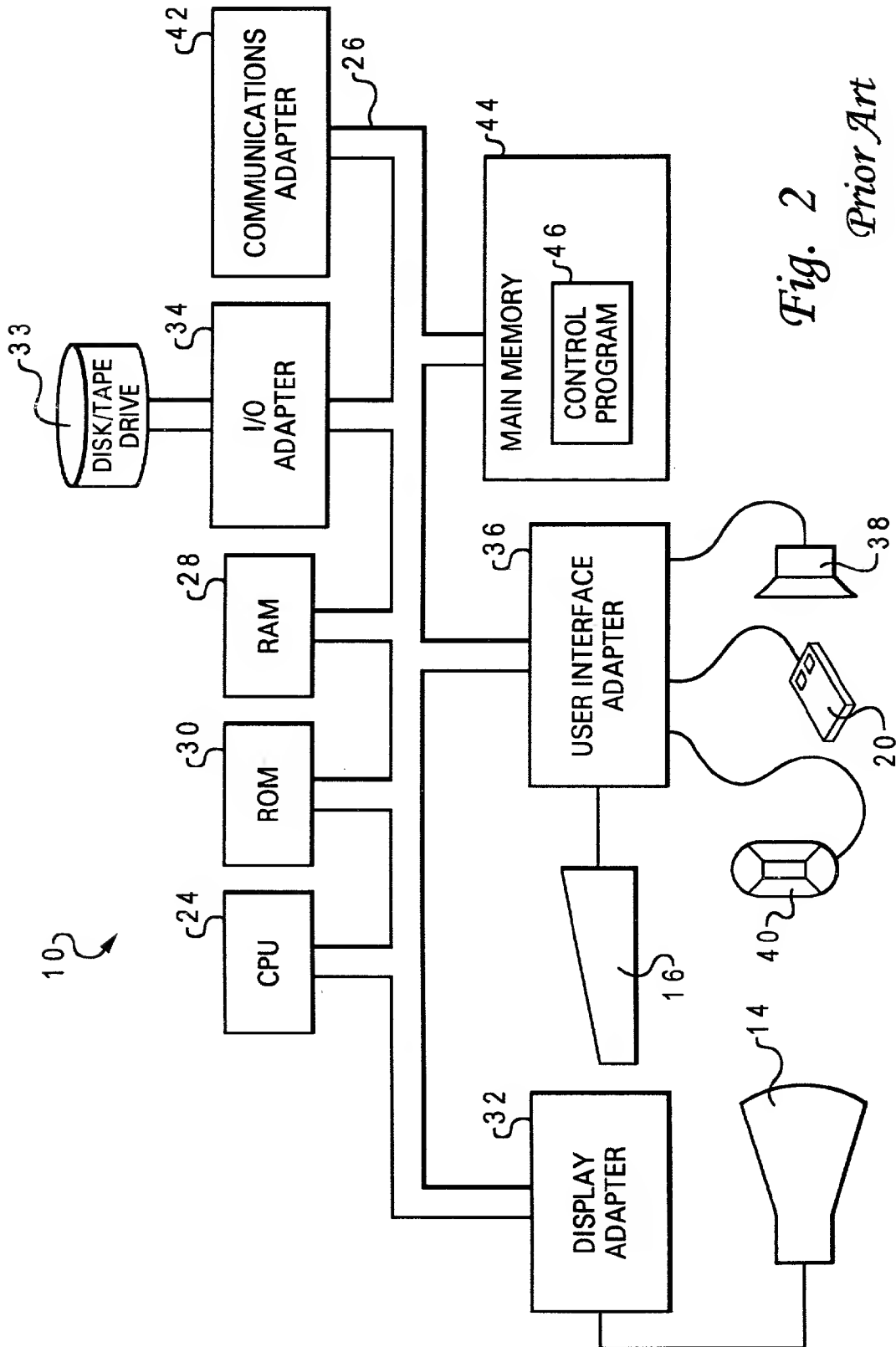
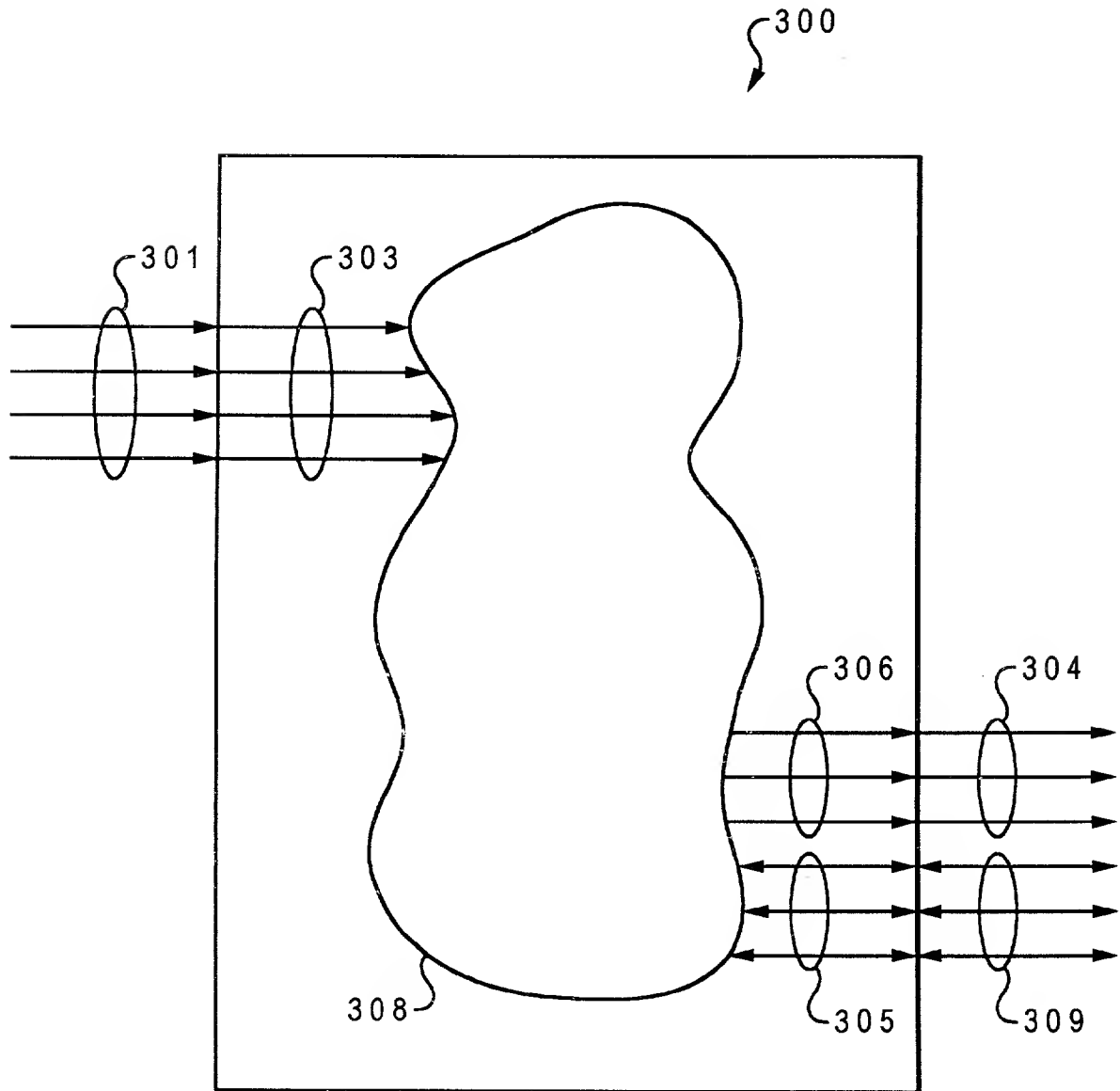


Fig. 2  
Prior Art

3/62



*Fig. 3A*

4/62

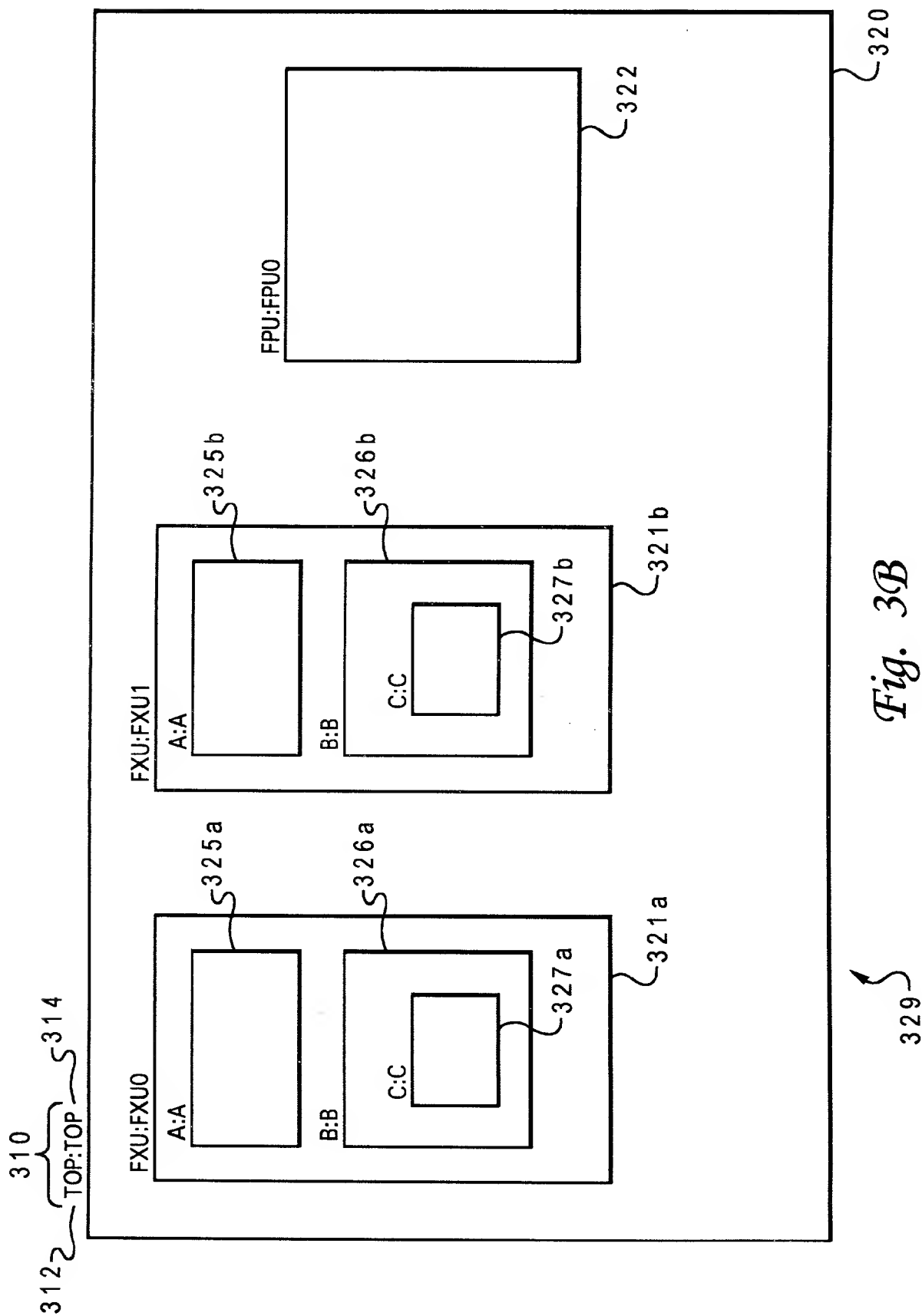
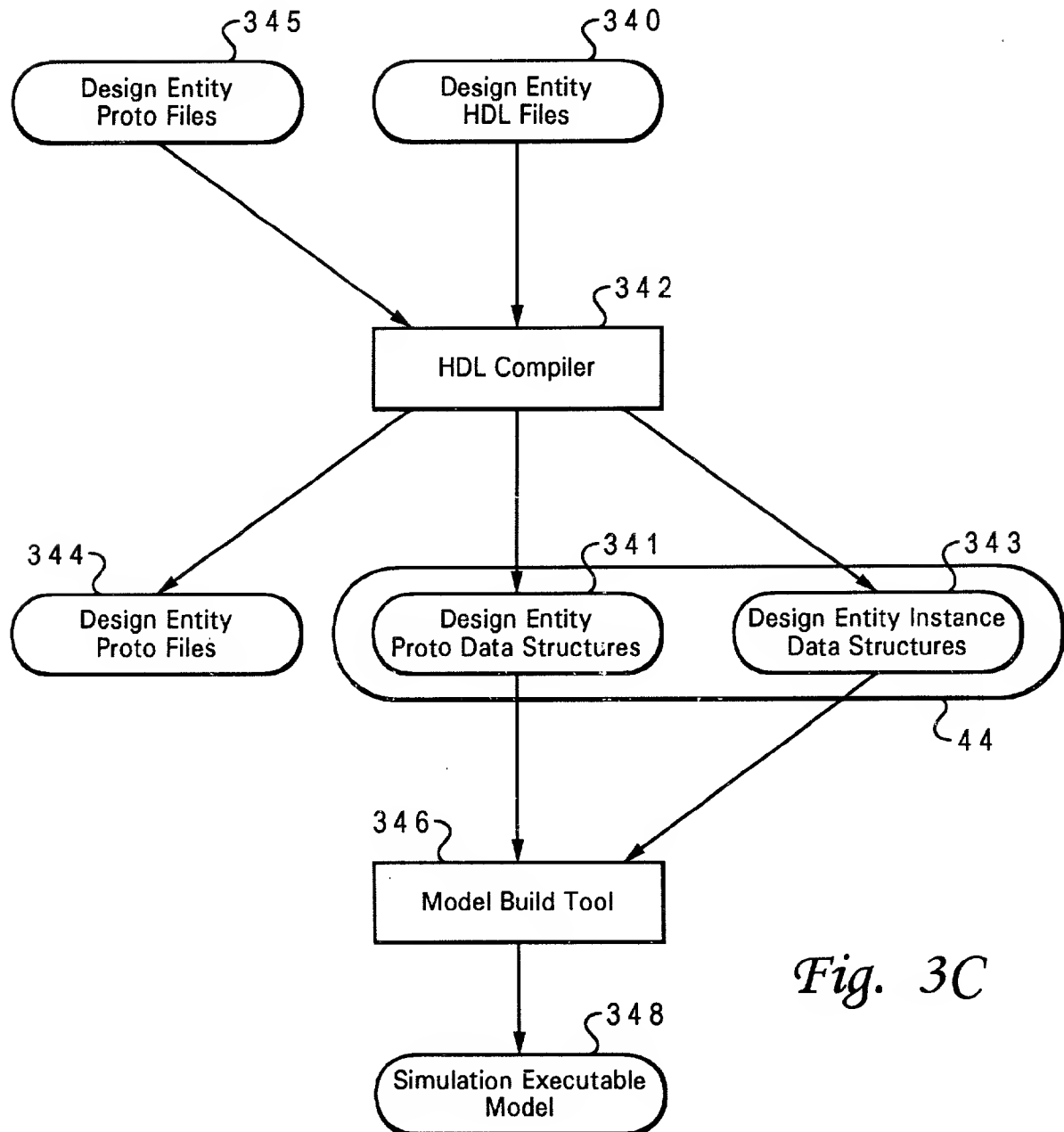


Fig. 3B

5/62

*Fig. 3C*

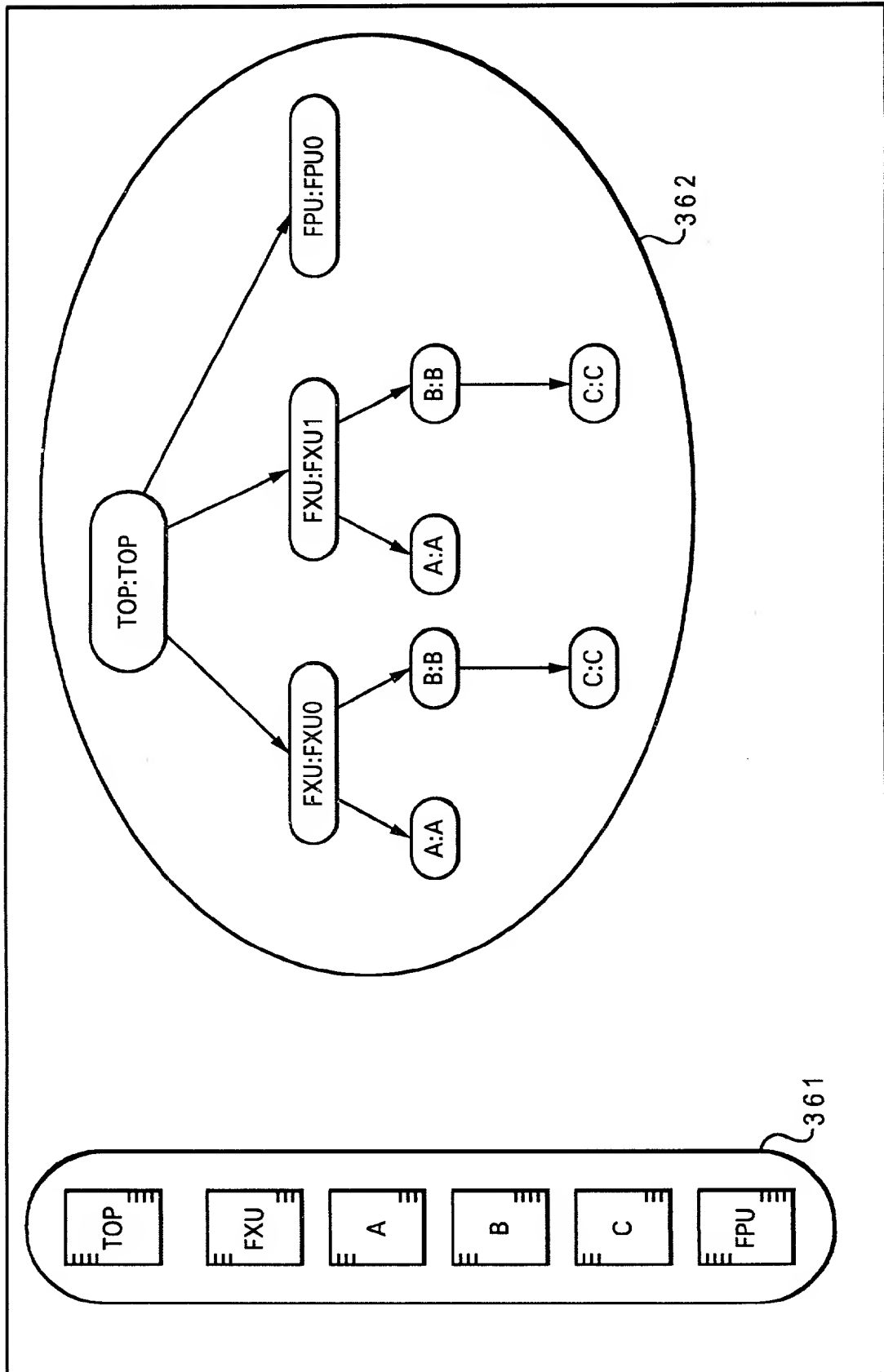


Fig. 3D

7/62

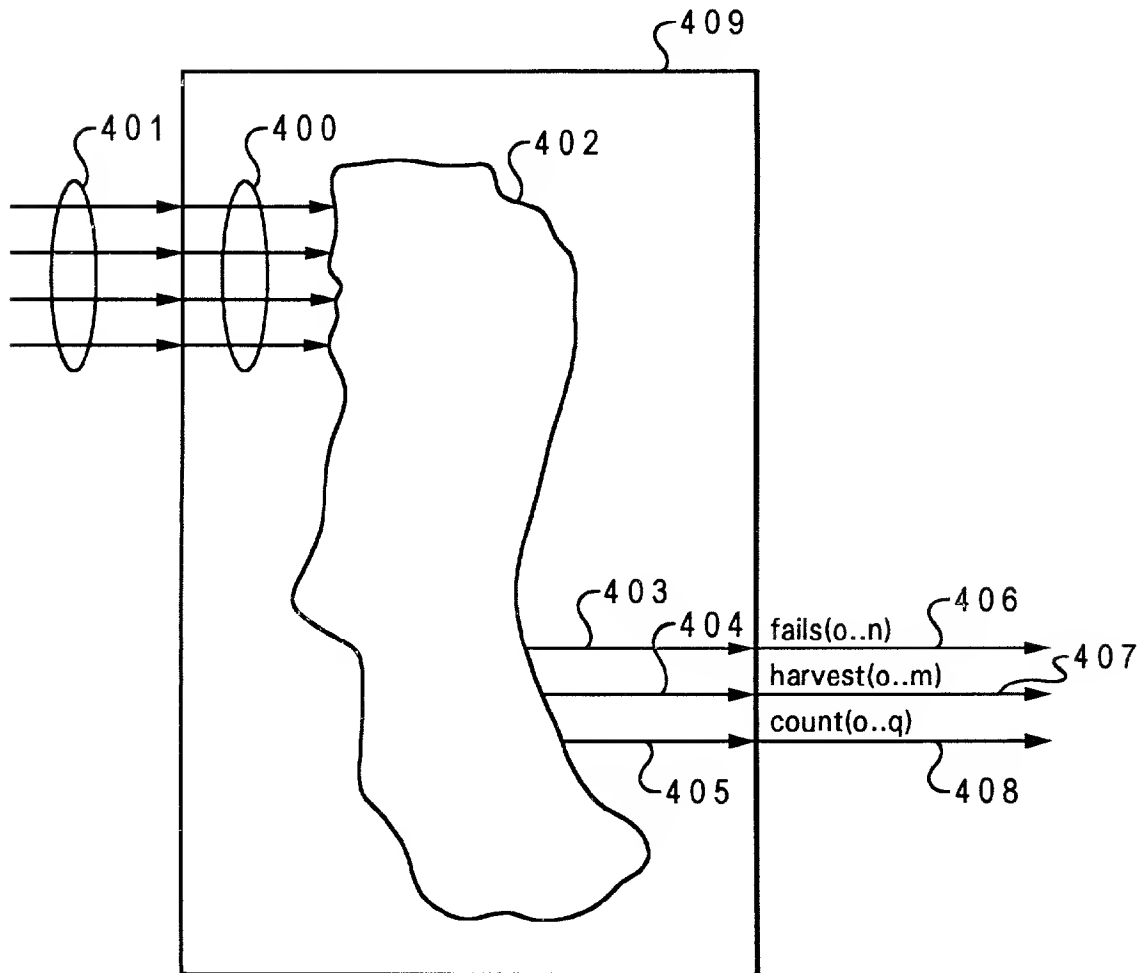


Fig. 4A

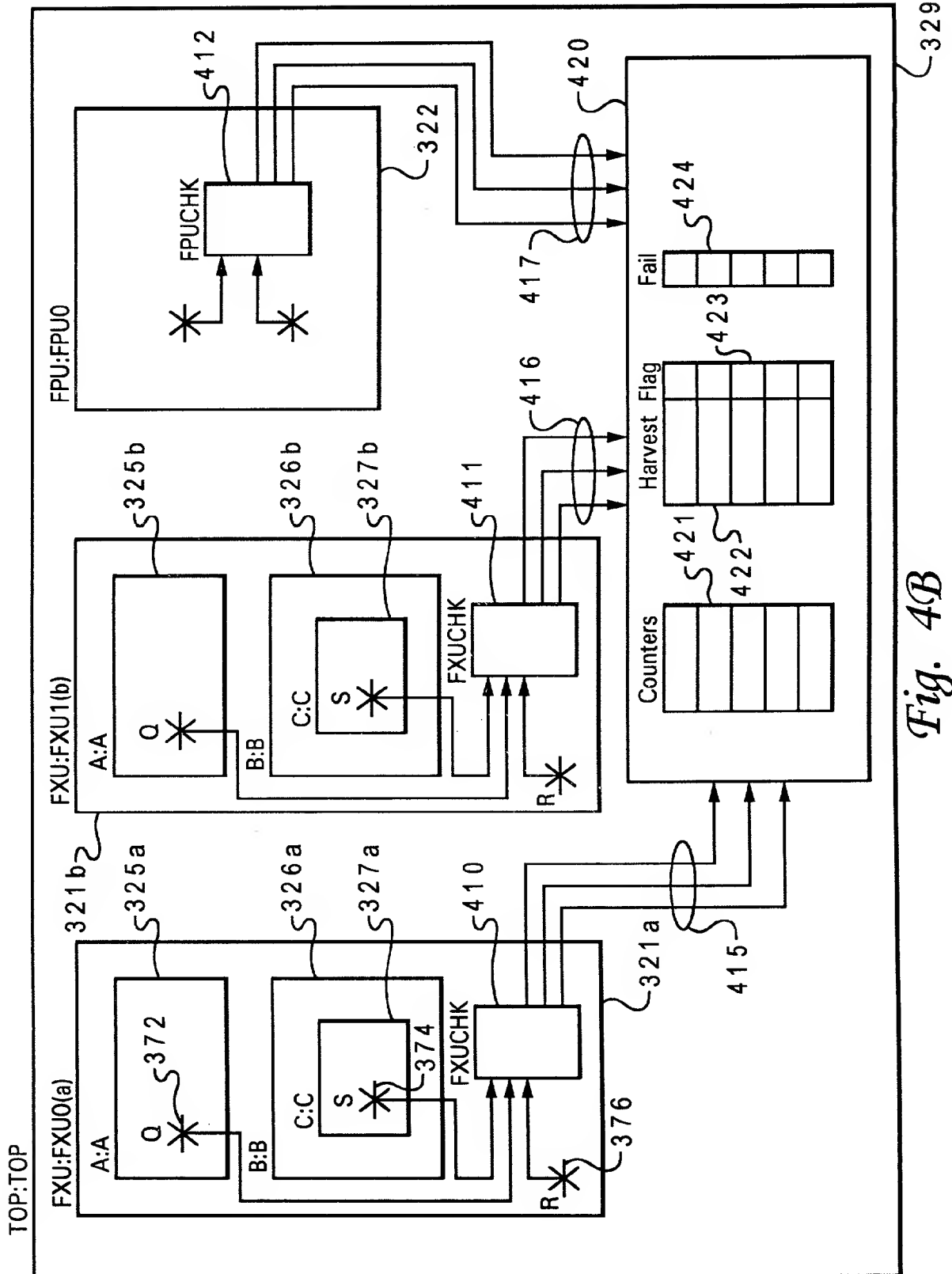


Fig. 4B



9/62

ENTITY FXUCHK IS

```

PORT(  S_IN      :    IN std_ulogic;
        Q_IN      :    IN std_ulogic;
        R_IN      :    IN std_ulogic;
        clock     :    IN std_ulogic;
        fails      :    OUT std_ulogic_vector(0 to 1);
        counts     :    OUT std_ulogic_vector(0 to 2);
        harvests   :    OUT std_ulogic_vector(0 to 1);
);

```

4 5 0

4 5 2 { --!! BEGIN  
--!! Design Entity: FXU;

4 5 3 { --!! Inputs  
--!! S\_IN => B.C.S;  
--!! Q\_IN => A.Q;  
--!! R\_IN => R;  
--!! CLOCK => clock;  
--!! End Inputs

4 5 4 { --!! Fail Outputs;  
--!! 0 : "Fail message for failure event 0";  
--!! 1 : "Fail message for failure event 1";  
--!! End Fail Outputs;

4 5 5 { --!! Count Outputs;  
--!! 0 : <event0> clock;  
--!! 1 : <event1> clock;  
--!! 2 : <event2> clock;  
--!! End Count Outputs;

4 5 6 { --!! Harvest Outputs;  
--!! 0 : "Message for harvest event 0";  
--!! 1 : "Message for harvest event 1";  
--!! End Harvest Outputs;

4 5 7 { --!! End;

4 5 1

4 4 0

ARCHITECTURE example of FXUCHK IS

BEGIN

... HDL code for entity body section ...

END;

4 5 8

*Fig. 4C*

10/62

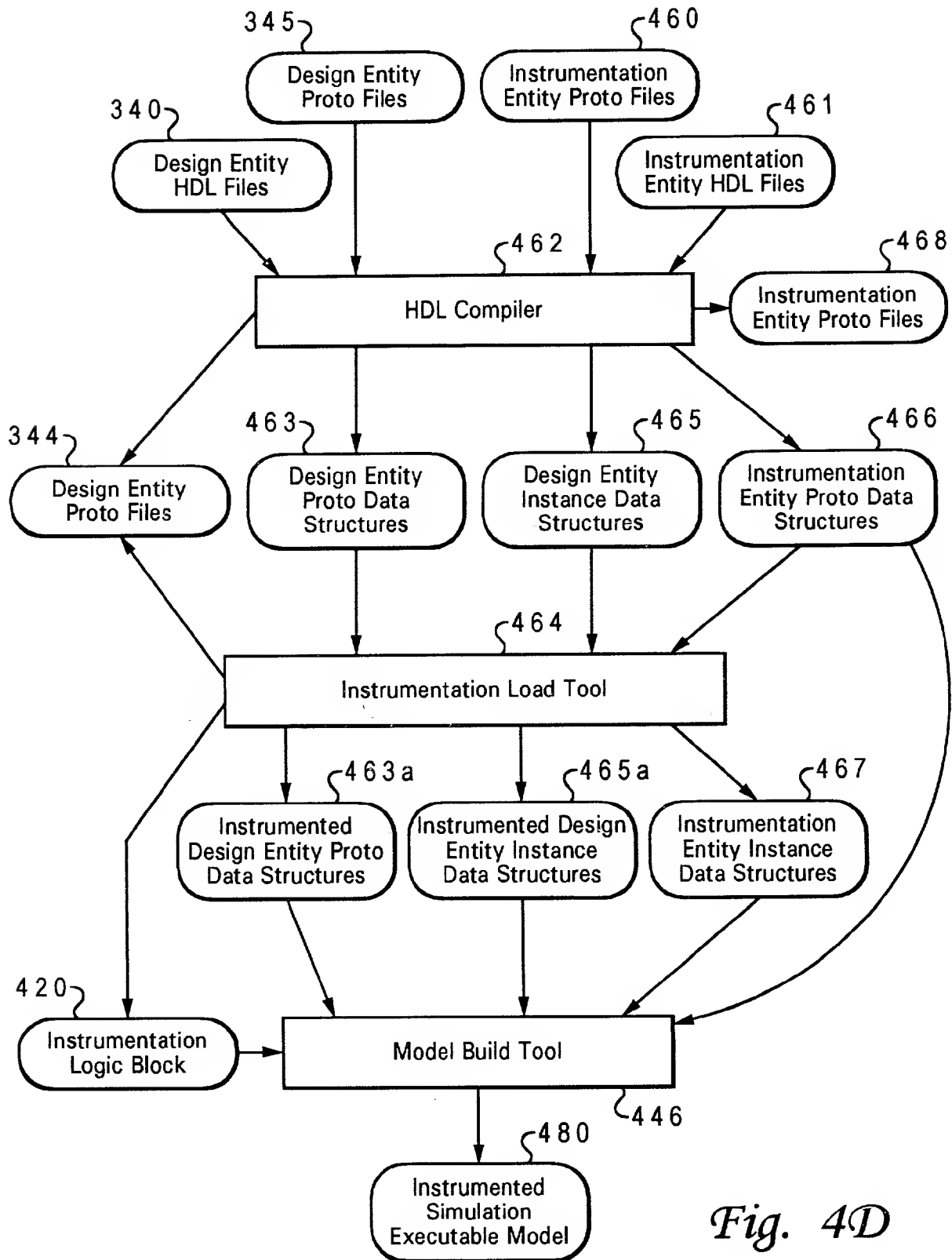


Fig. 4D

11/62

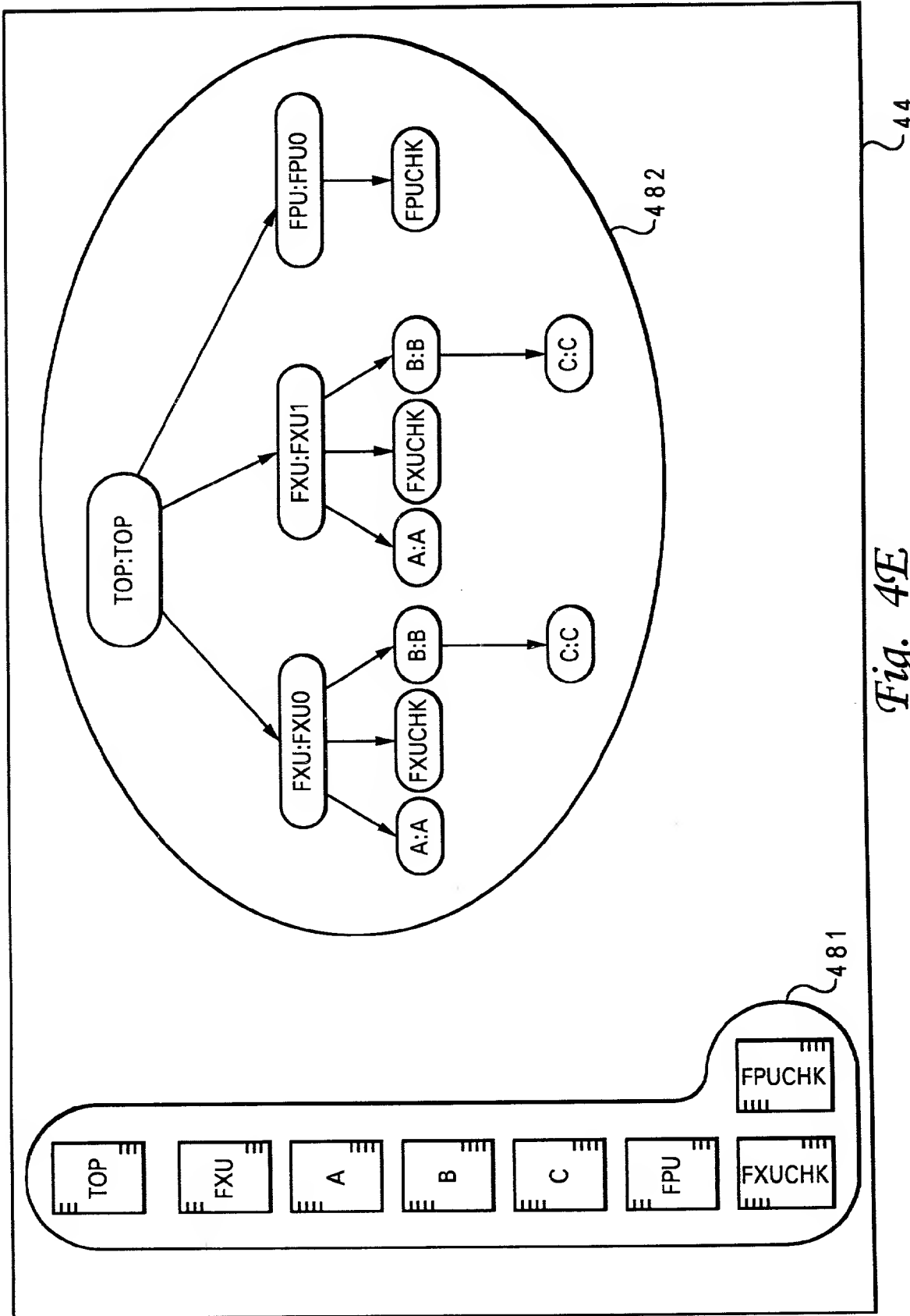
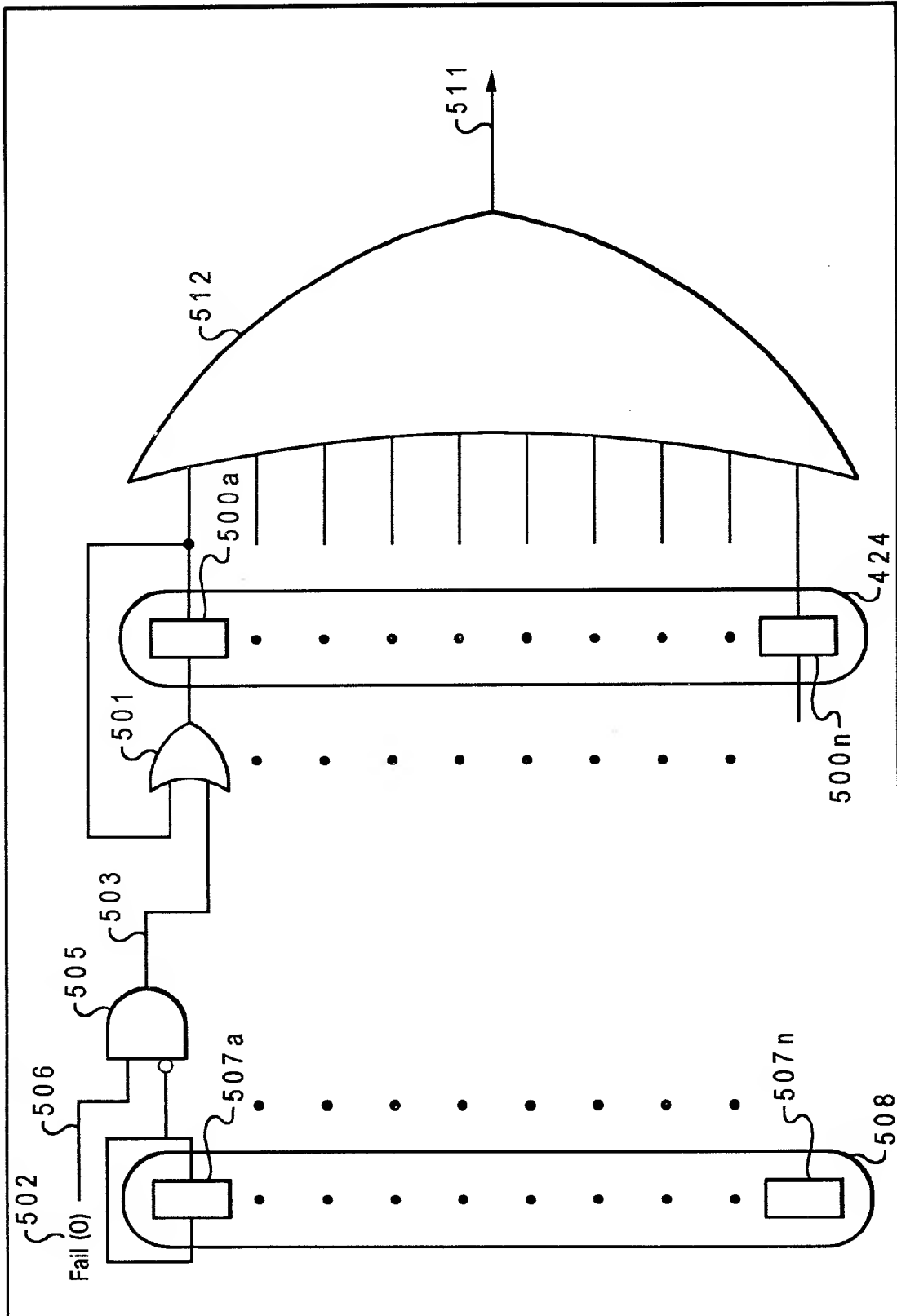


Fig. 4E



13/62

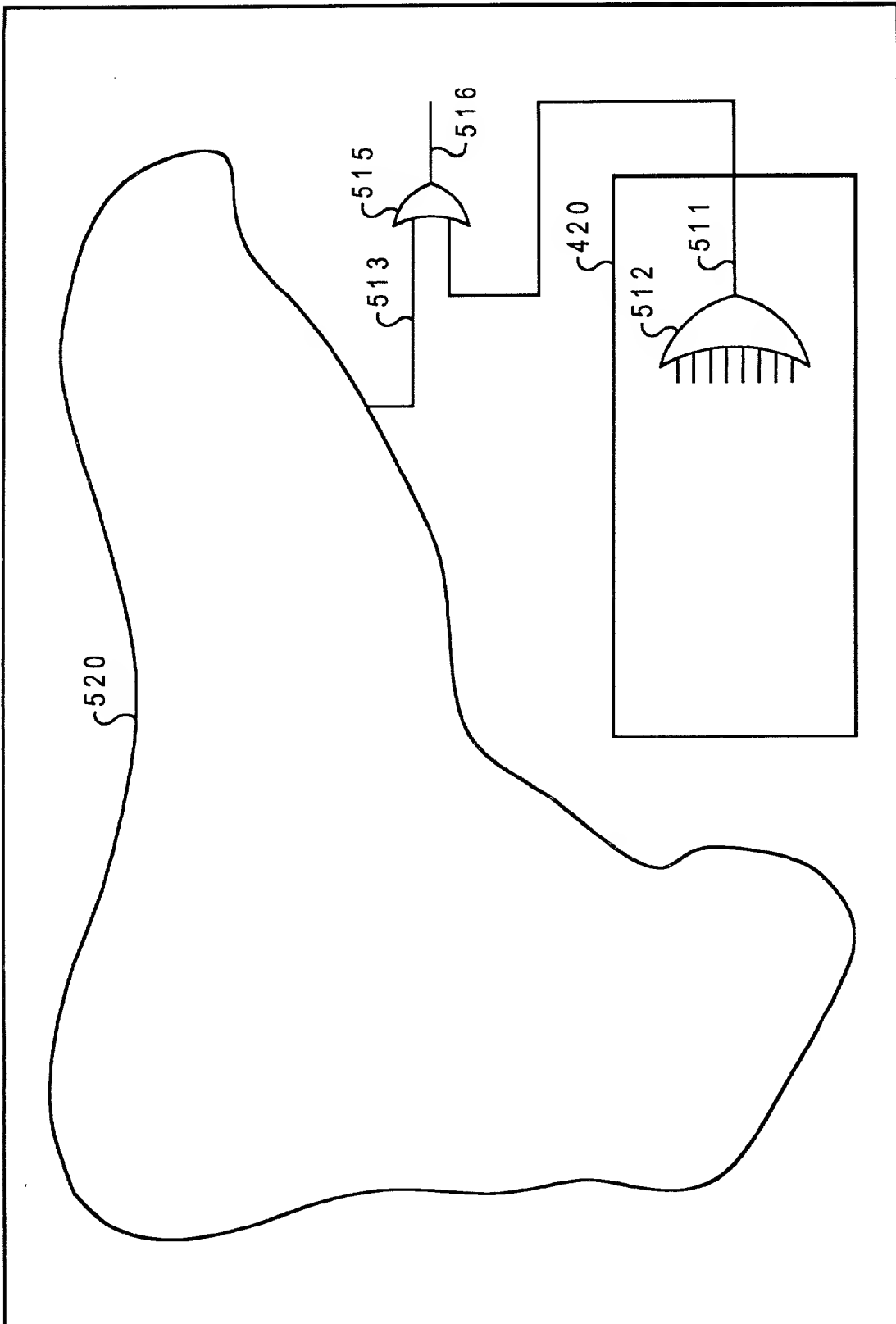


Fig. 5B

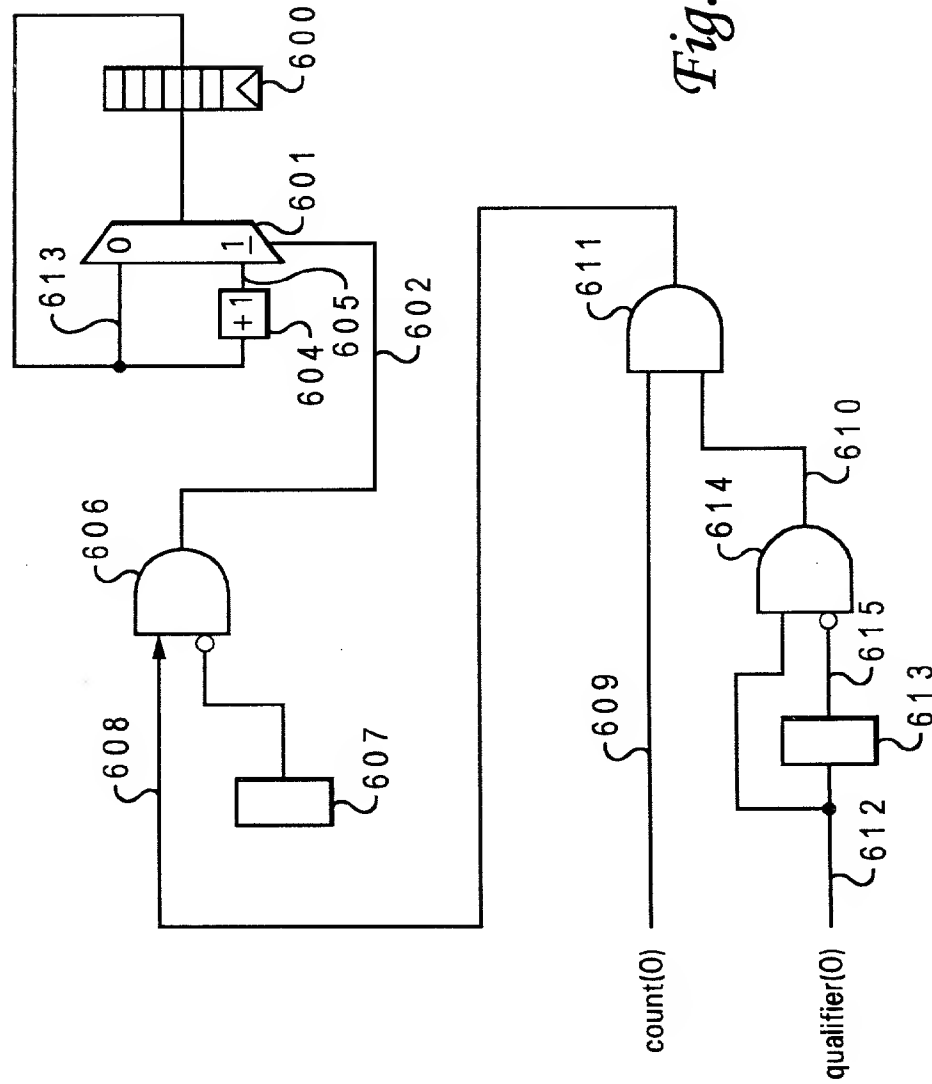


Fig. 6A

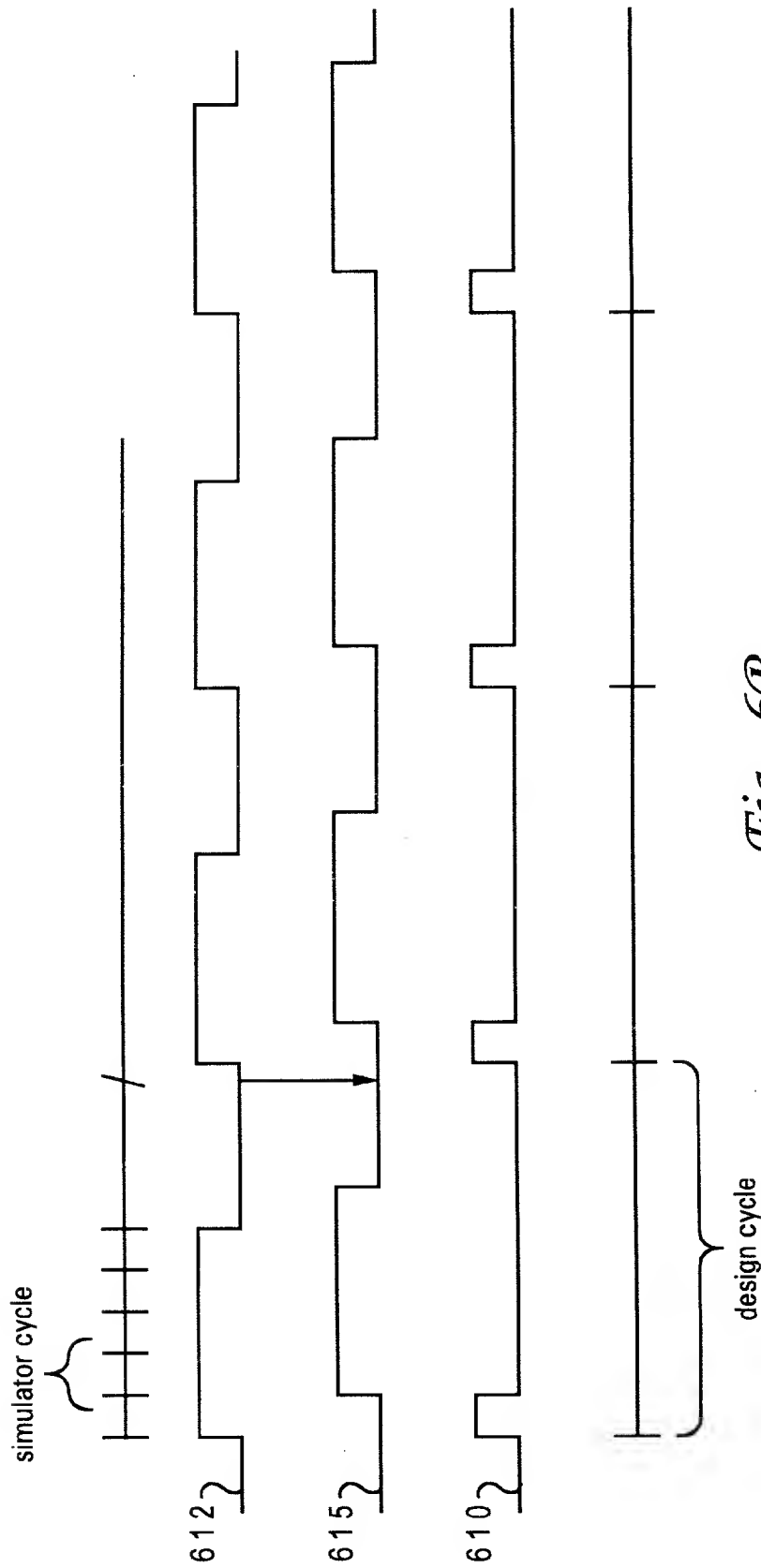


Fig. 6B

16/62

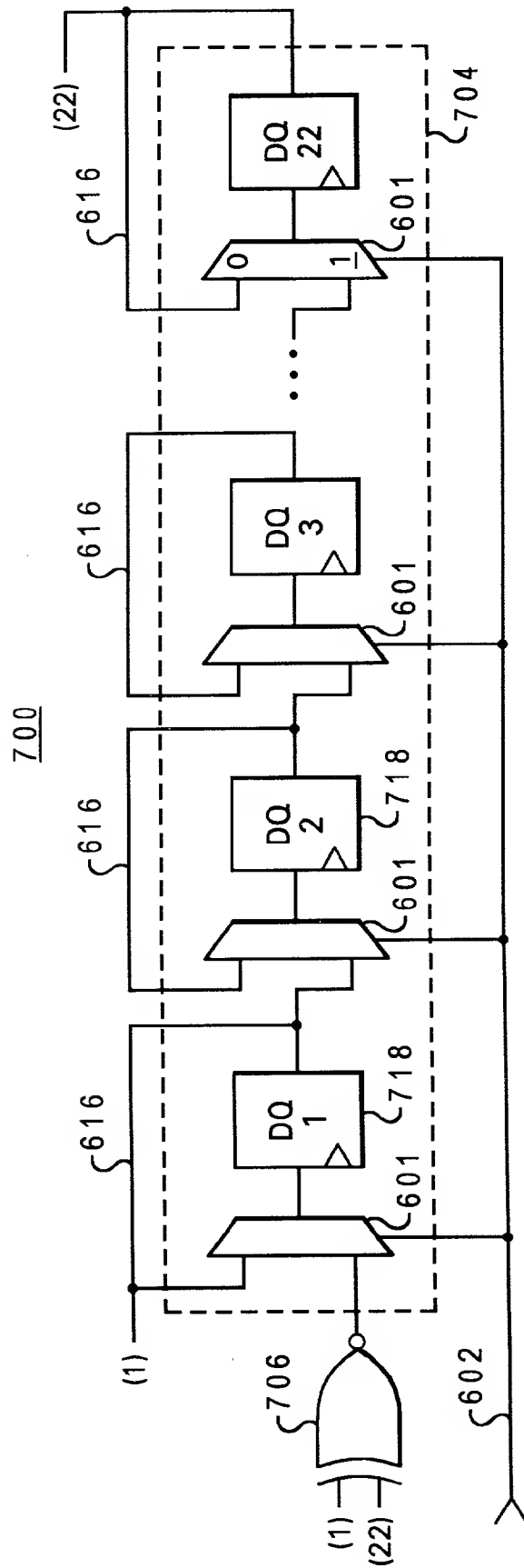
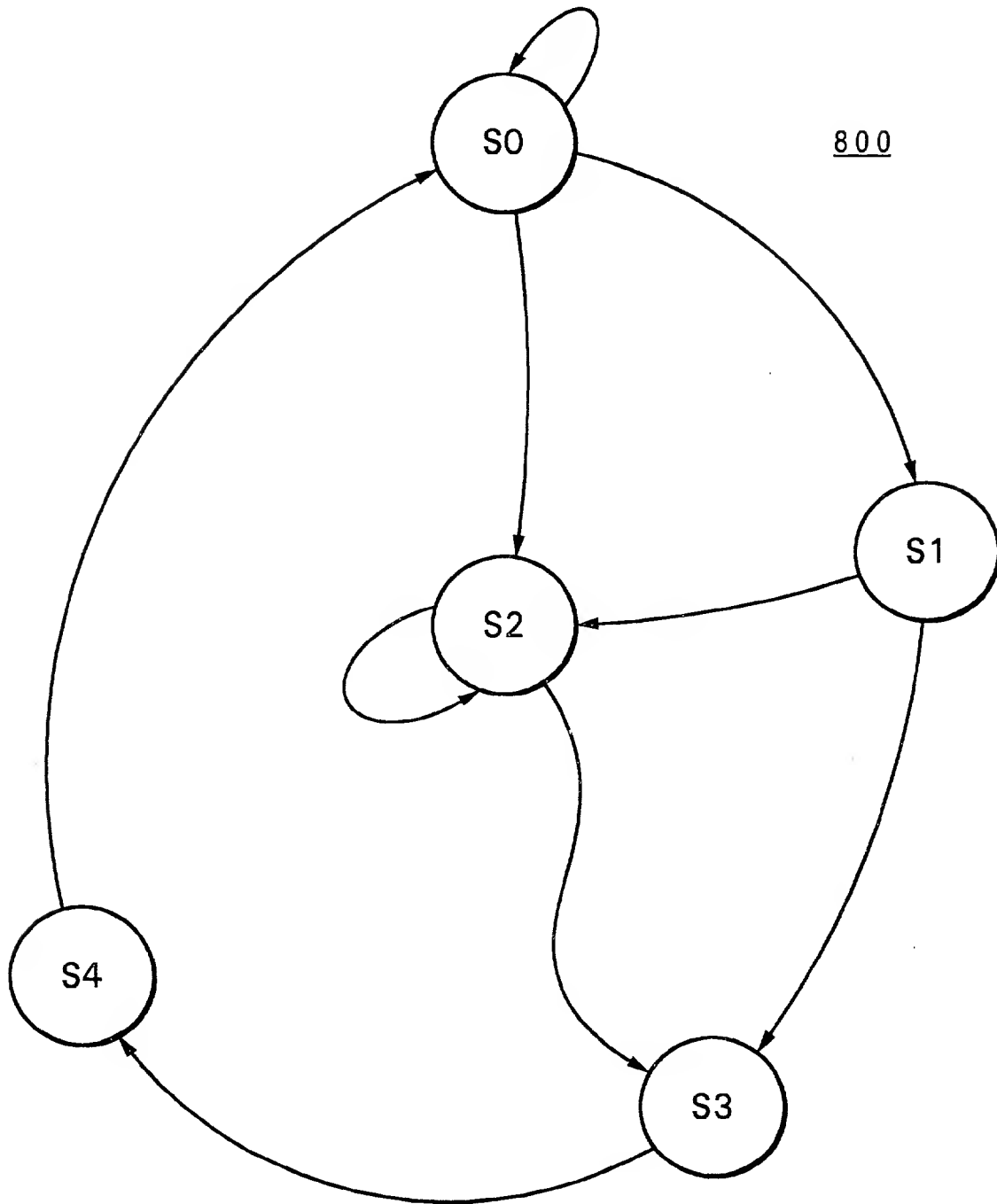


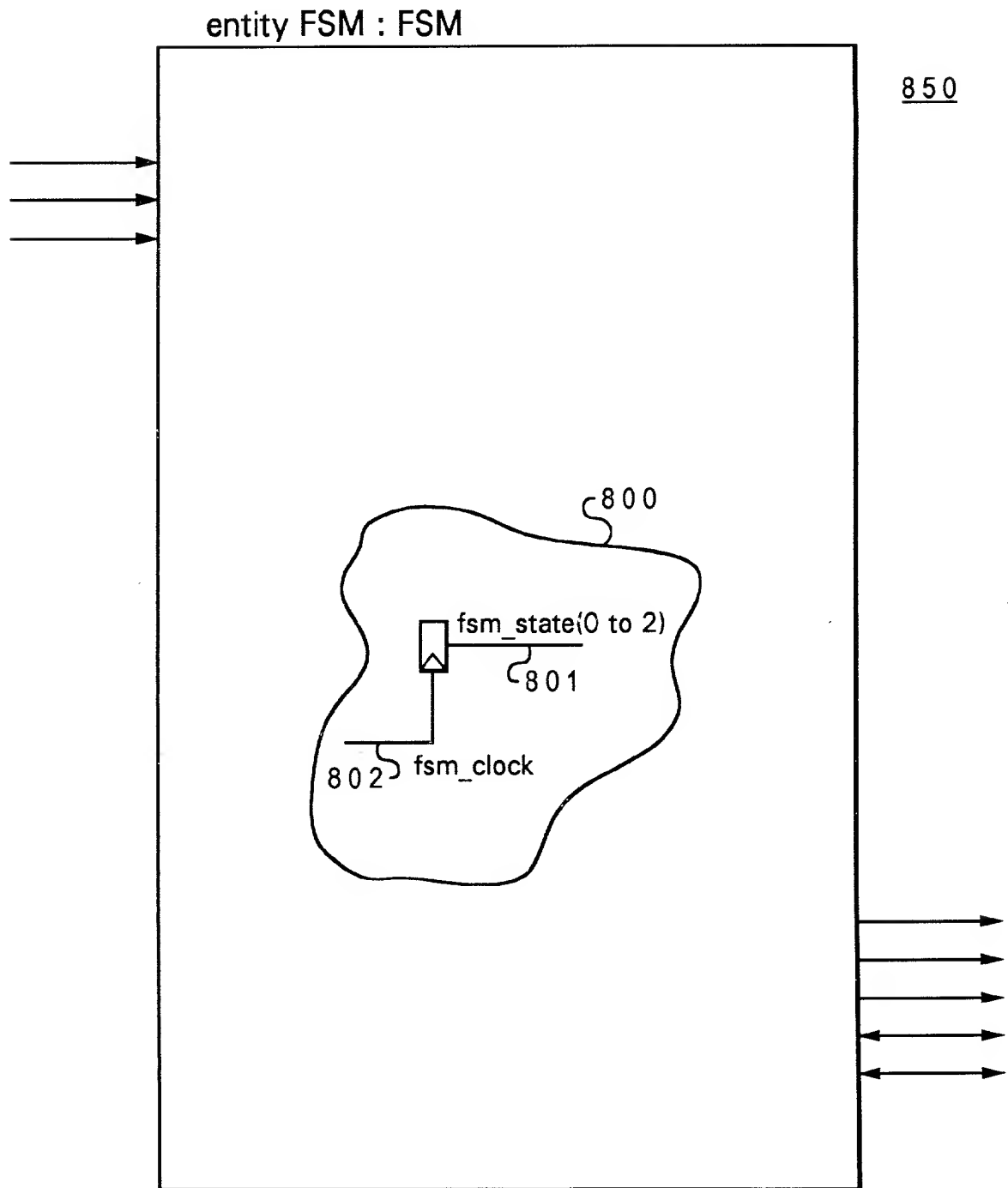
Fig. 7





*Fig. 8A*

*Prior Art*

*Fig. 8B**Prior Art*

19/62

ENTITY FSM IS

```
PORT(
    ....ports for entity fsm....
);
```

ARCHITECTURE FSM OF FSM IS

BEGIN

... HDL code for FSM and rest of the entity ...

fsm\_state(0 to 2) <= ... Signal 801 ...

8 5 3	{	--!! Embedded FSM : examplefsm;			
8 5 9	{	--!! clock : (fsm_clock);			
8 5 4	{	--!! state_vector : (fsm_state(0 to 2));			
8 5 5	{	--!! states : (S0, S1, S2, S3, S4);			
8 5 6	{	--!! state_encoding : ('000', '001', '010', '011', '100');			
8 5 7	{	--!! arcs : (S0 => S0, S0 => S1, S0 => S2,	} 8 5 2	} 8 6 0	
	{	--!! (S1 => S2, S1 => S3, S2 => S2,			
	{	--!! (S2 => S3, S3 => S4, S4 => S0);			
8 5 8	{	--!! End FSM;			

END;

*Fig. 8C*

entity FSM : FSM

850

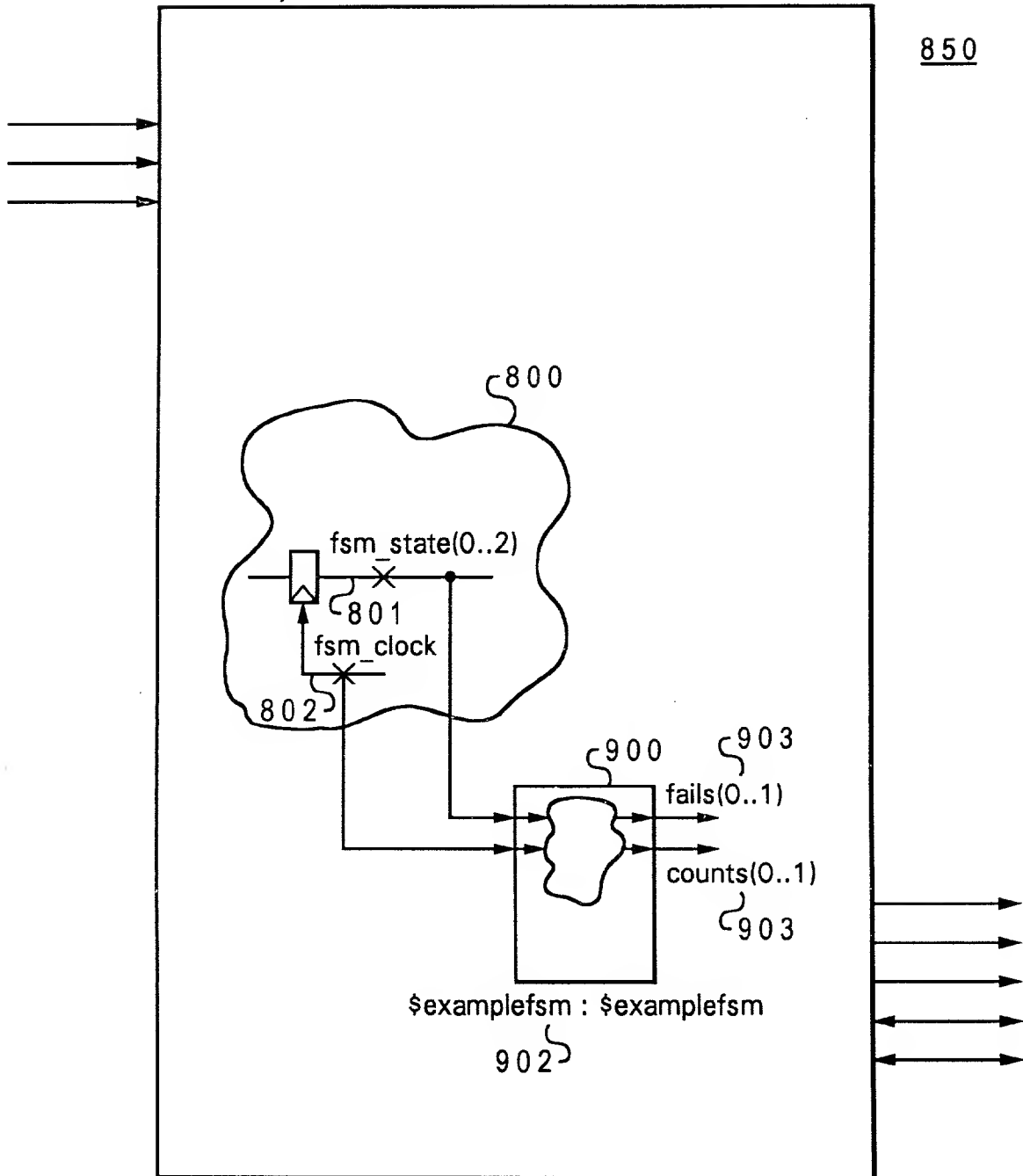
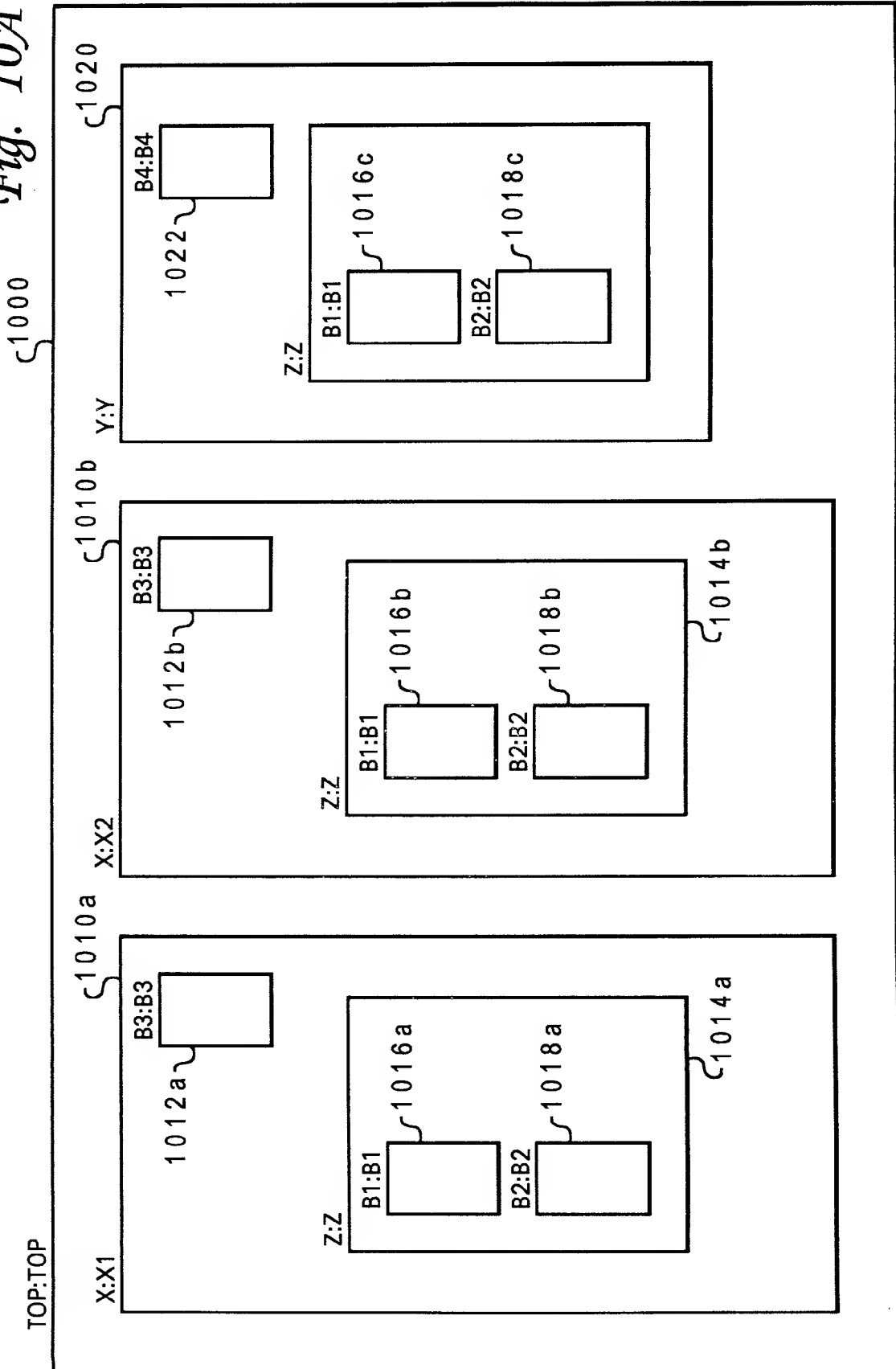


Fig. 9

21/62

Fig. 10A



1030 1032 1034 1036  
<instantiation identifier> . <instrumentation entity name> . <design entity name> . <eventname>

Fig. 10B

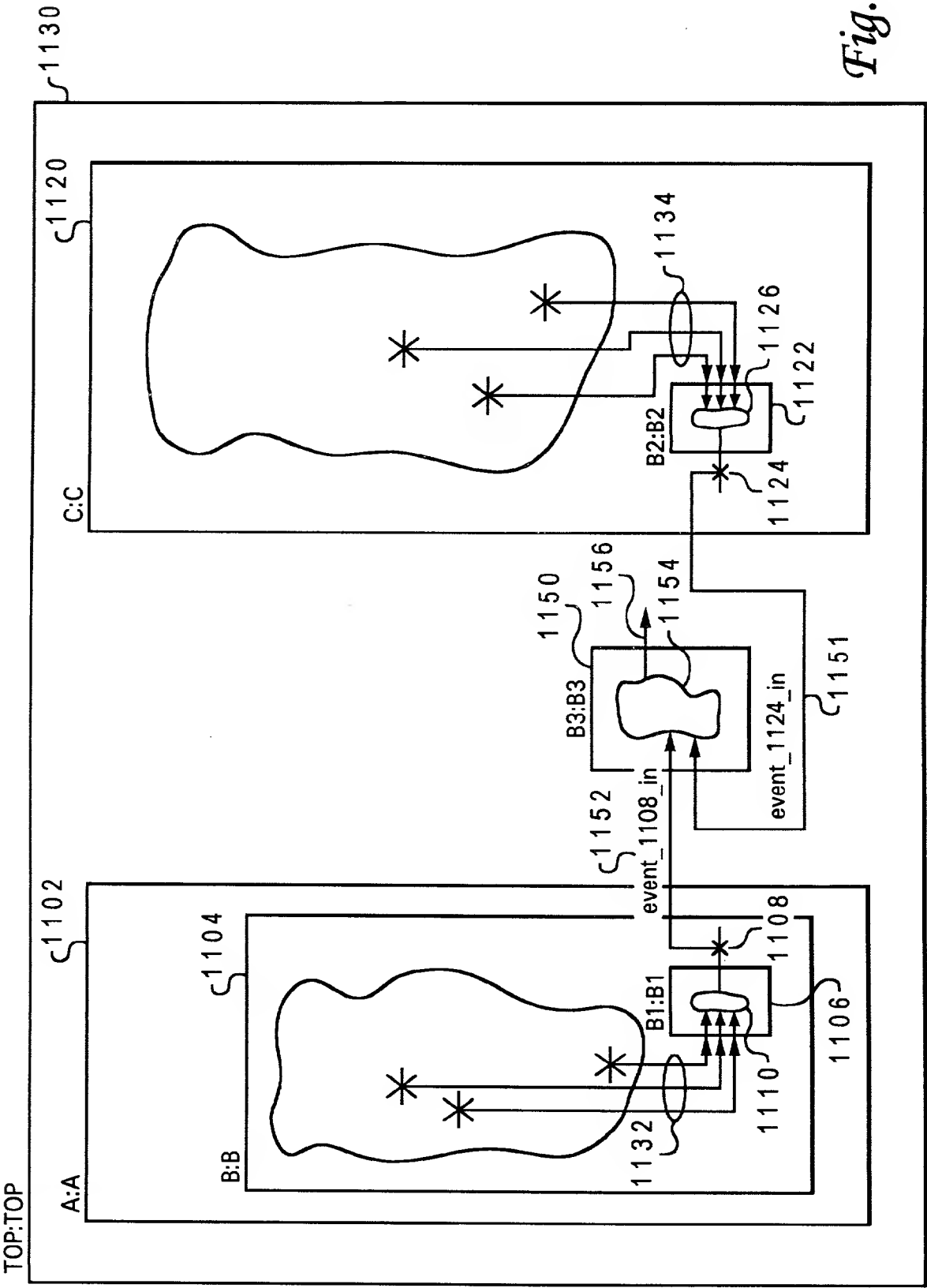
X1	B3	X	COUNT1	1040
X1.Z	B1	Z	COUNT1	1041
X1.Z	B2	Z	COUNT1	1042
X2	B3	X	COUNT1	1043
X2.Z	B1	Z	COUNT1	1044
X2.Z	B2	Z	COUNT1	1045
Y	B4	Y	COUNT1	1046
Y.Z	B1	Z	COUNT1	1047
Y.Z	B2	Z	COUNT1	1048

Fig. 10C

1030 1034 1036  
<instantiation identifier> . <design entity name> . <eventname>

Fig. 10D

Fig. 11A



24/62

--!! Inputs  
--!! event\_1108\_in <= C.[B2.count.event\_1108];  
--!! event\_1124\_in <= A.B.[B1.count.event\_1124];  
--!! End Inputs

1163 1165 1161 1162 1164 1166

The diagram shows two event input lines. The first line, 'event\_1108\_in', is associated with count '1163' and event number '1165'. The second line, 'event\_1124\_in', is associated with count '1164' and event number '1166'. Both lines are grouped by a bracket labeled '1161'. The event numbers '1161' and '1162' are also associated with the second line.

*Fig. 11B*

--!! Inputs  
--!! event\_1108\_in <= C.[count.event\_1108];  
--!! event\_1124\_in <= B.[count.event\_1124];  
--!! End Inputs

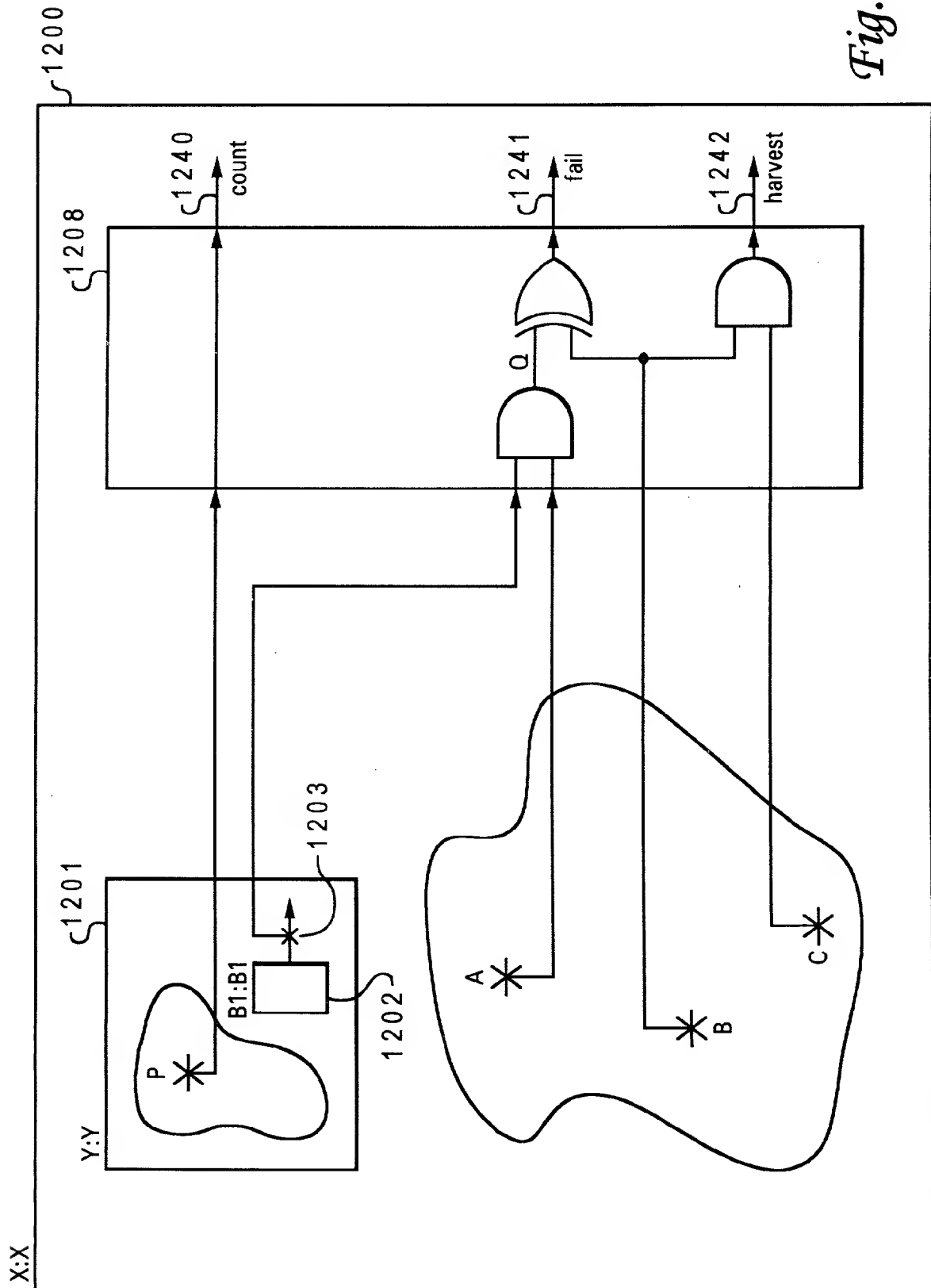
1171 1172

The diagram shows two event input lines. The first line, 'event\_1108\_in', is associated with count '1171' and event number '1172'. The second line, 'event\_1124\_in', is associated with count '1172' and event number '1171'.

*Fig. 11C*



*Fig. 12A*



ENTITY X IS

PORT( :  
:  
:  
);

ARCHITECTURE example of X IS

BEGIN

.  
.  
.  
.  
... HDL code for X ...  
.  
.  
.  
.

1 2 2 1 { Y:Y  
PORT MAP( :  
:  
);

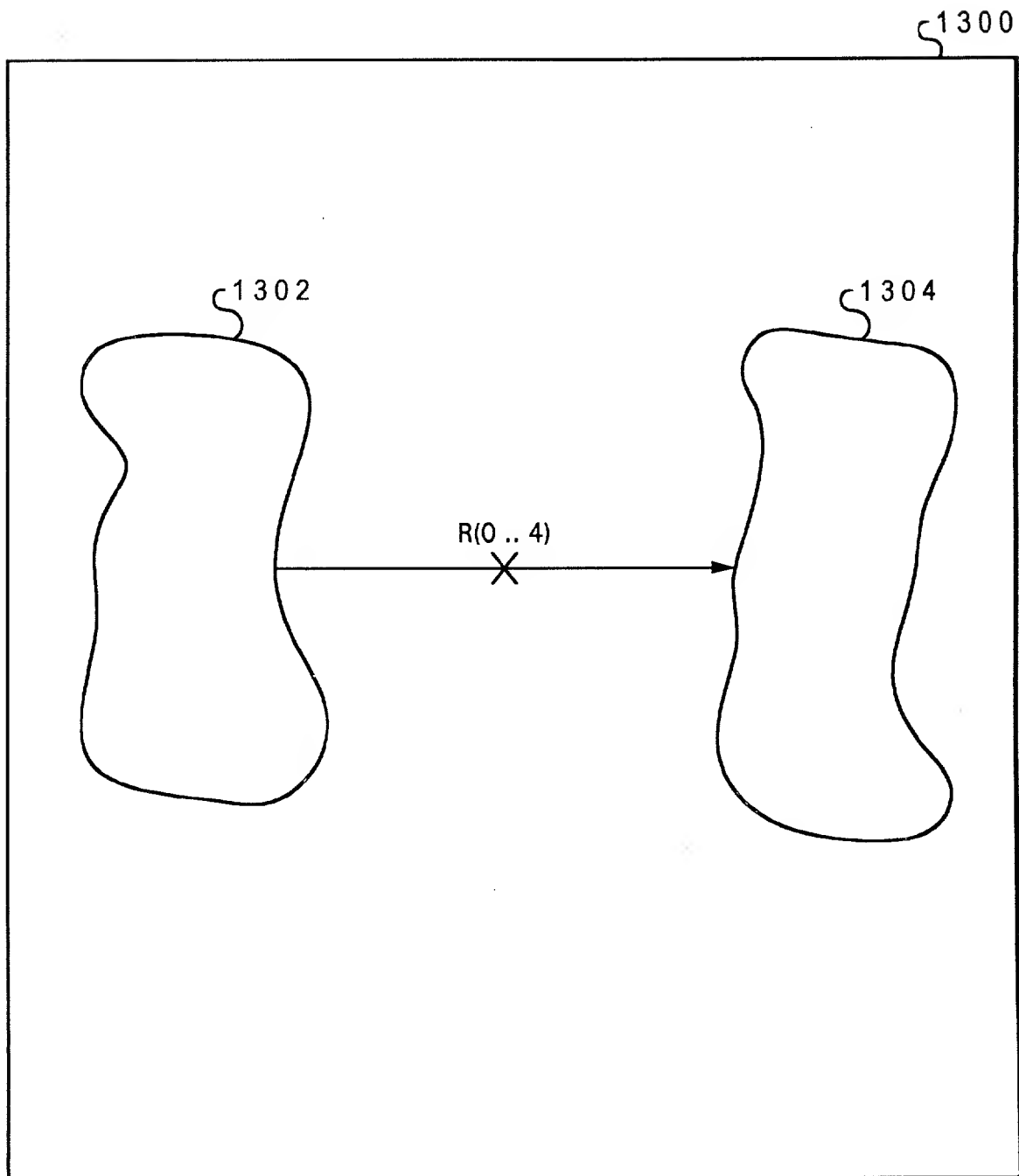
1 2 2 2 { A <= ....  
B <= ....  
C <= ....

1 2 2 3 { --!! [count, countname0, clock] <= Y.P; 1 2 3 0  
--!! Q <= Y. [B1.count.count1] AND A; 1 2 3 2  
--!! [fail, failname0, "fail msg"] <= Q XOR B; 1 2 3 4  
--!! [harvest, harvestname0, "harvest msg"] <= B AND C;  
END; 1 2 3 6

1 2 2 0

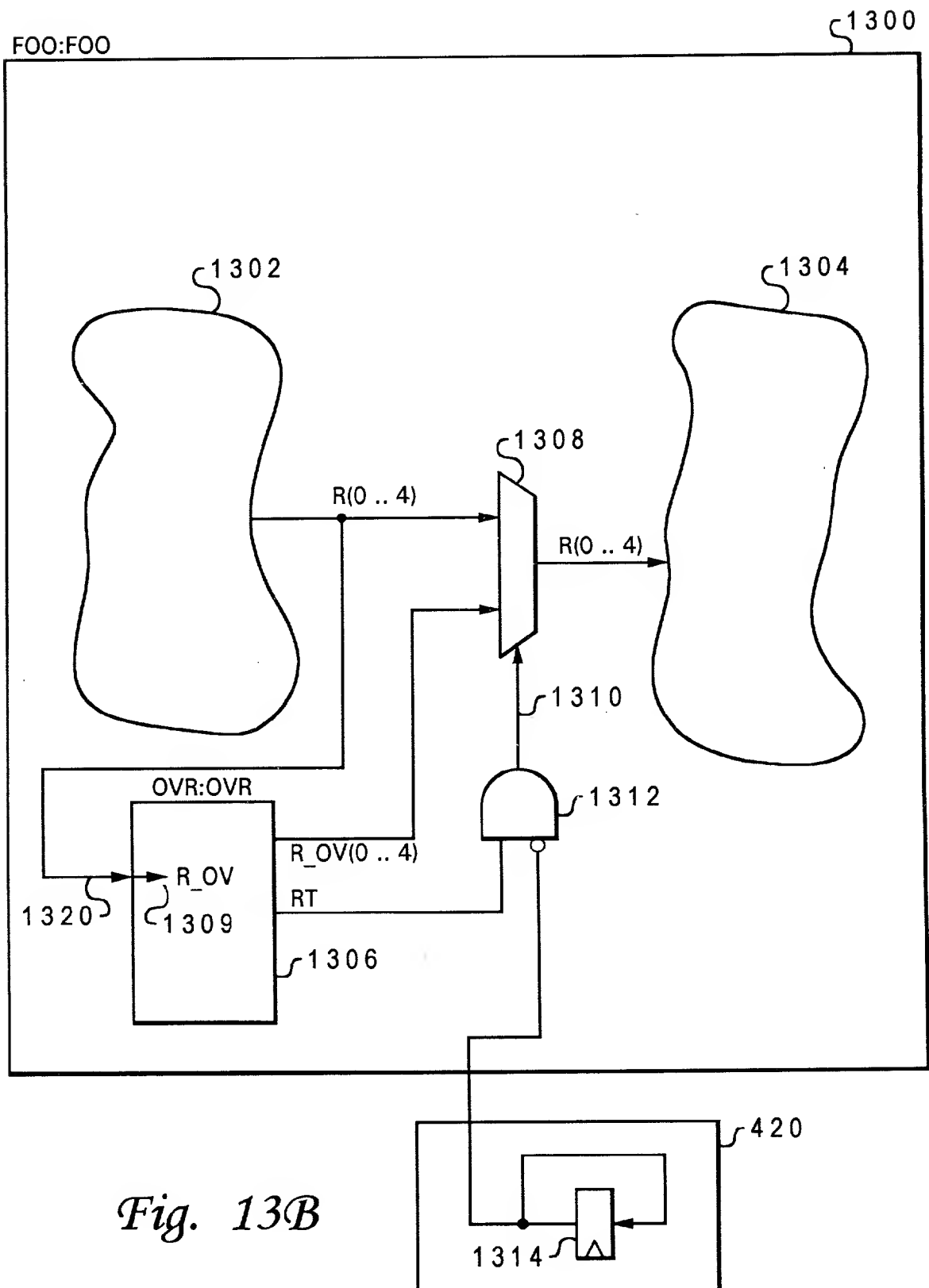
*Fig. 12B*

27/62



*Fig. 13A*

28/62



29/62

```

ENTITY OVR IS
    PORT(  R_IN      :  IN std_ulogic_vector(0 .. 4);
          .
          .
          ... other ports as required ...
          .
          .
          R_OV      :  OUT std_ulogic_vector(0 .. 4);
          RT        :  OUT std_ulogic
    );

--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)
--!! R_IN => {R(0 .. 4)};
--!! :
... other ports as needed ...
--!! :
--!! End Inputs

--!! Outputs
--!! <R_OVRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS
    BEGIN
        ... HDL code for entity body section ...
    END;
    
```

Diagram annotations (brackets and labels):

- 1364: Bracket for the first PORT declaration.
- 1362: Bracket for the R\_OV and RT output declarations.
- 1363: Bracket for the BEGIN and Design Entity lines.
- 1360: Bracket for the R\_IN input declaration.
- 1361: Bracket for the Output section.
- 1351: Bracket for the End Inputs and End Outputs lines.
- 1356: Bracket for the entire entity body (from BEGIN to END).
- 1358: Bracket for the ARCHITECTURE example of OVR IS section.
- 1340: Large bracket on the right side encompassing the entire entity definition.

*Fig. 13C*

30/62

ENTITY FOO IS

PORT( :  
:  
:  
);

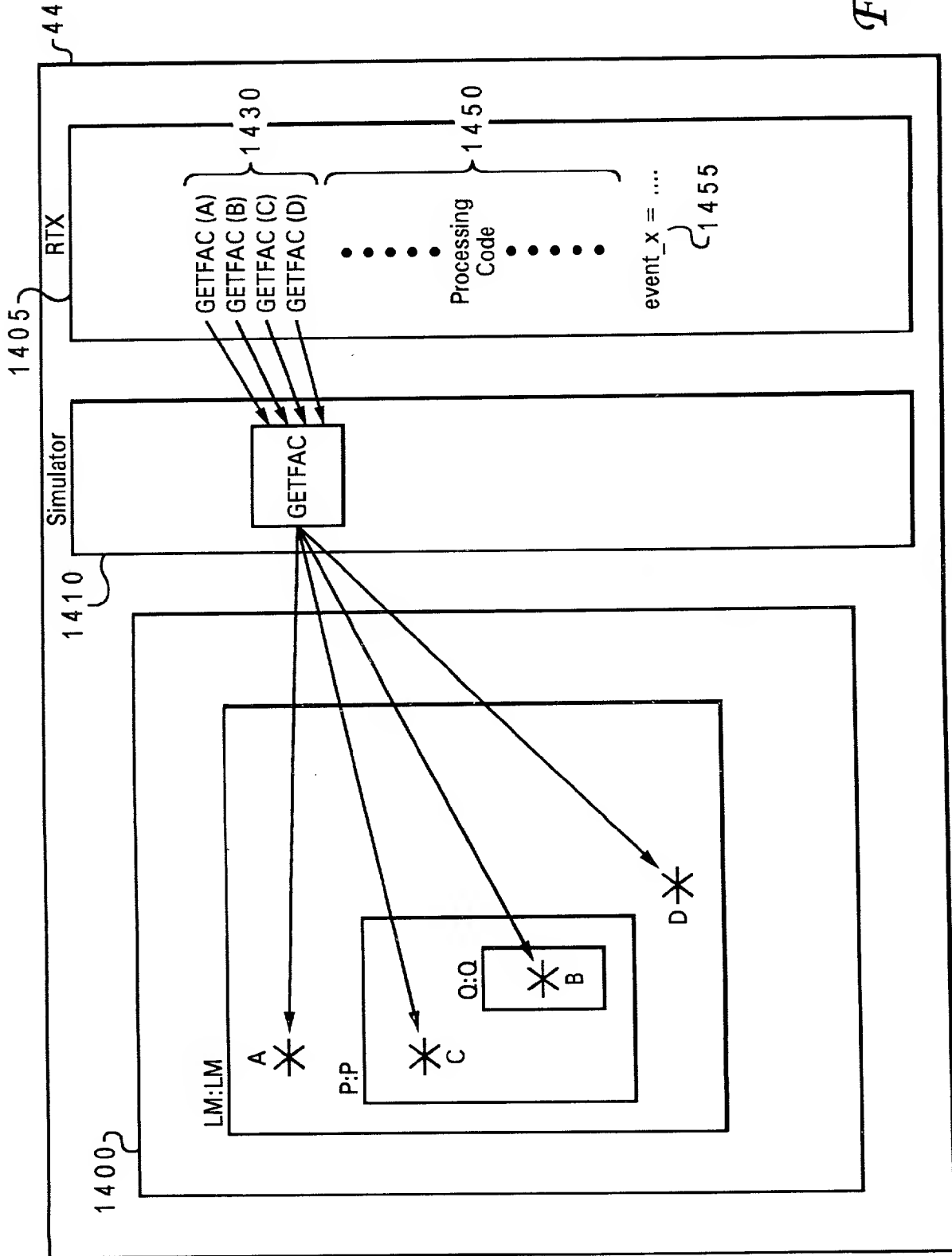
ARCHITECTURE example of FOO IS

BEGIN

.  
.  
.  
.  
.  
R <= .....  
.  
.  
.  
.  
.

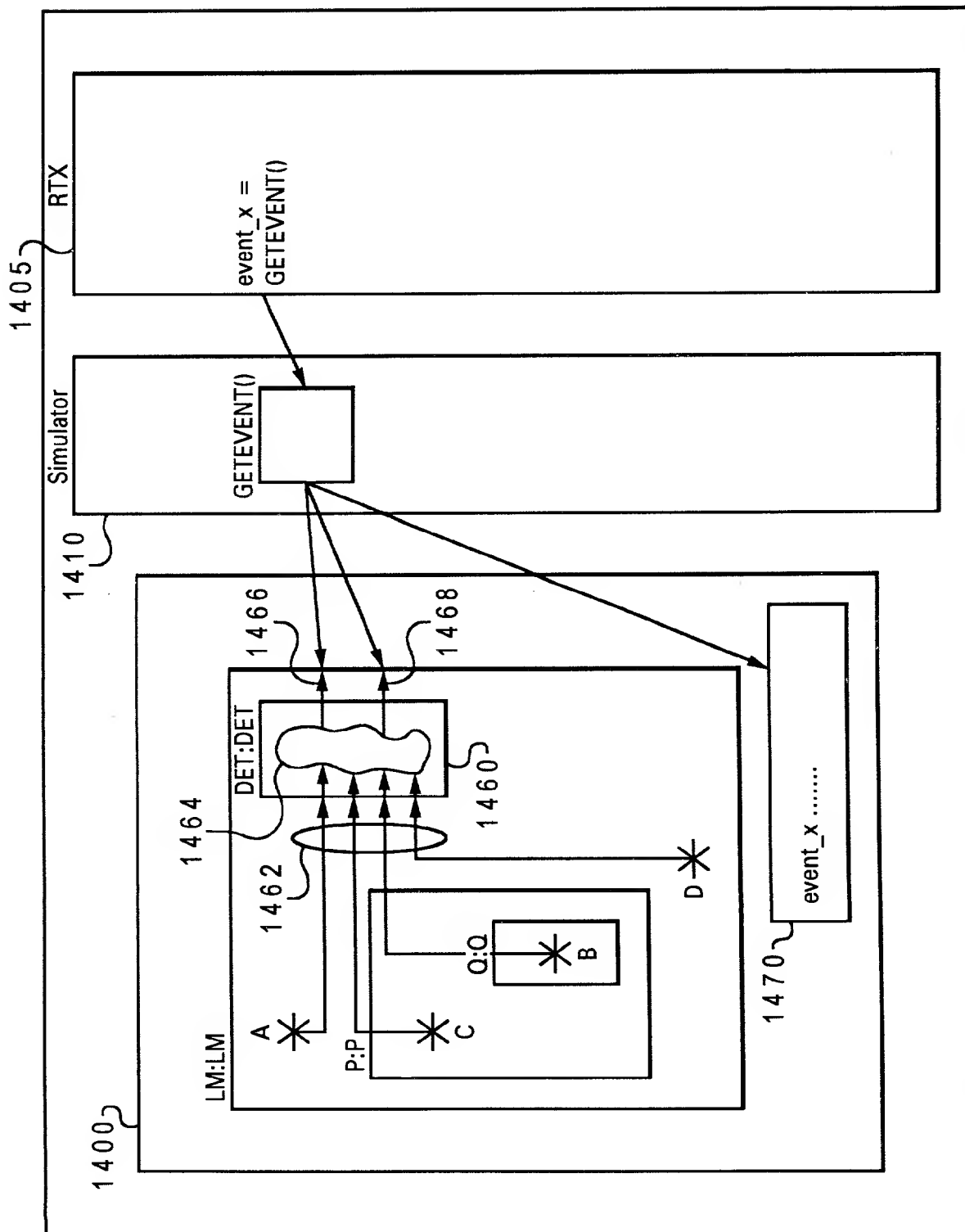
1380 { --!! R\_IN <= {R};  
--!!  
--!!  
--!! R\_OV(0 to 4) <= .....; 1383  
--!! RT <= .....; 1384  
--!! [override, R\_OVRRIDE, R(0 .. 4), RT] <= R\_OV(0 to 4);

Fig. 13D



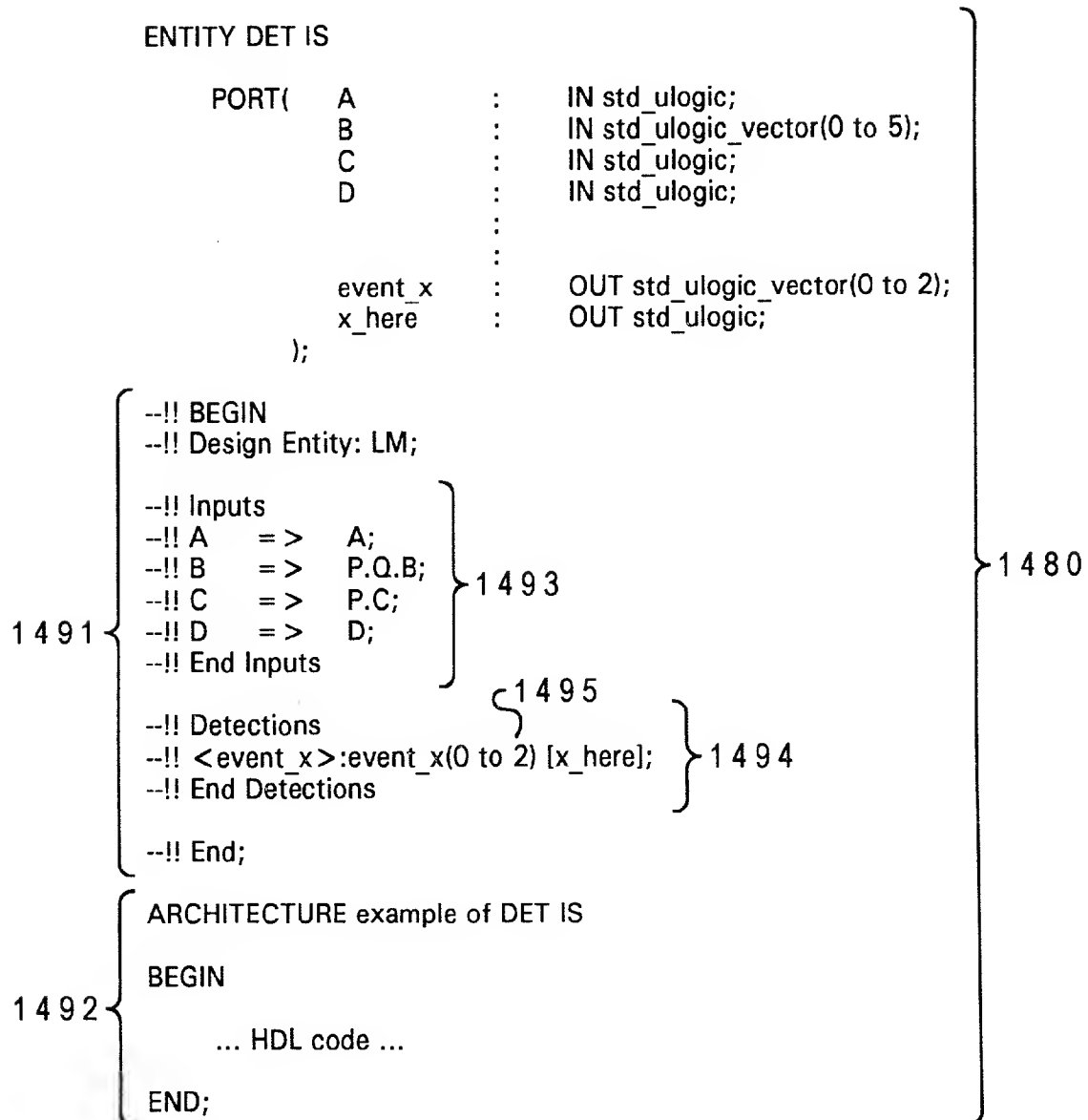
32/62

Fig. 14B



US 6,520,000 B2  
 Gabele, et al.  
 Tracking Converge Results In A Batch Simulation Farm Network



*Fig. 14C*

34/62

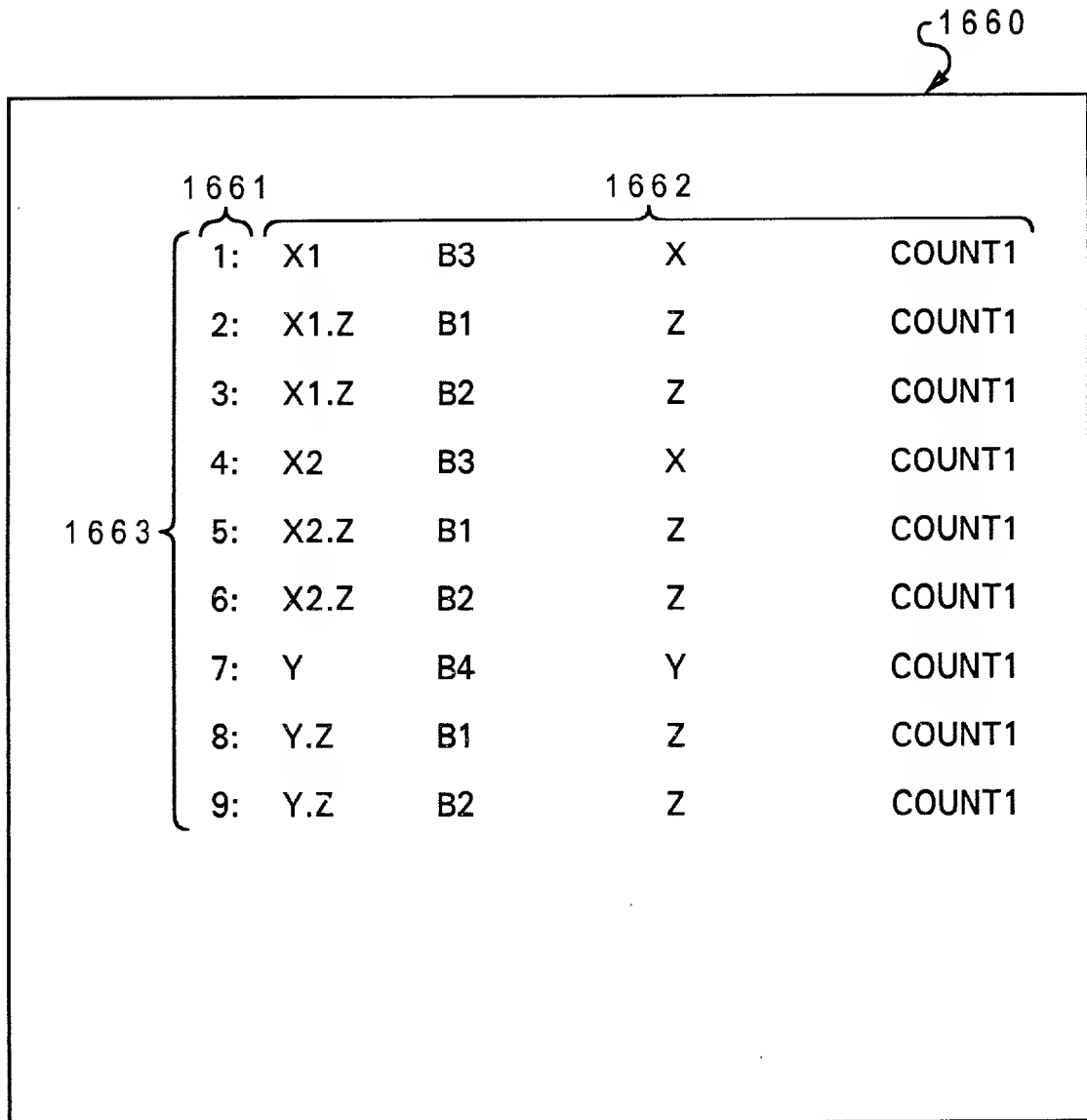
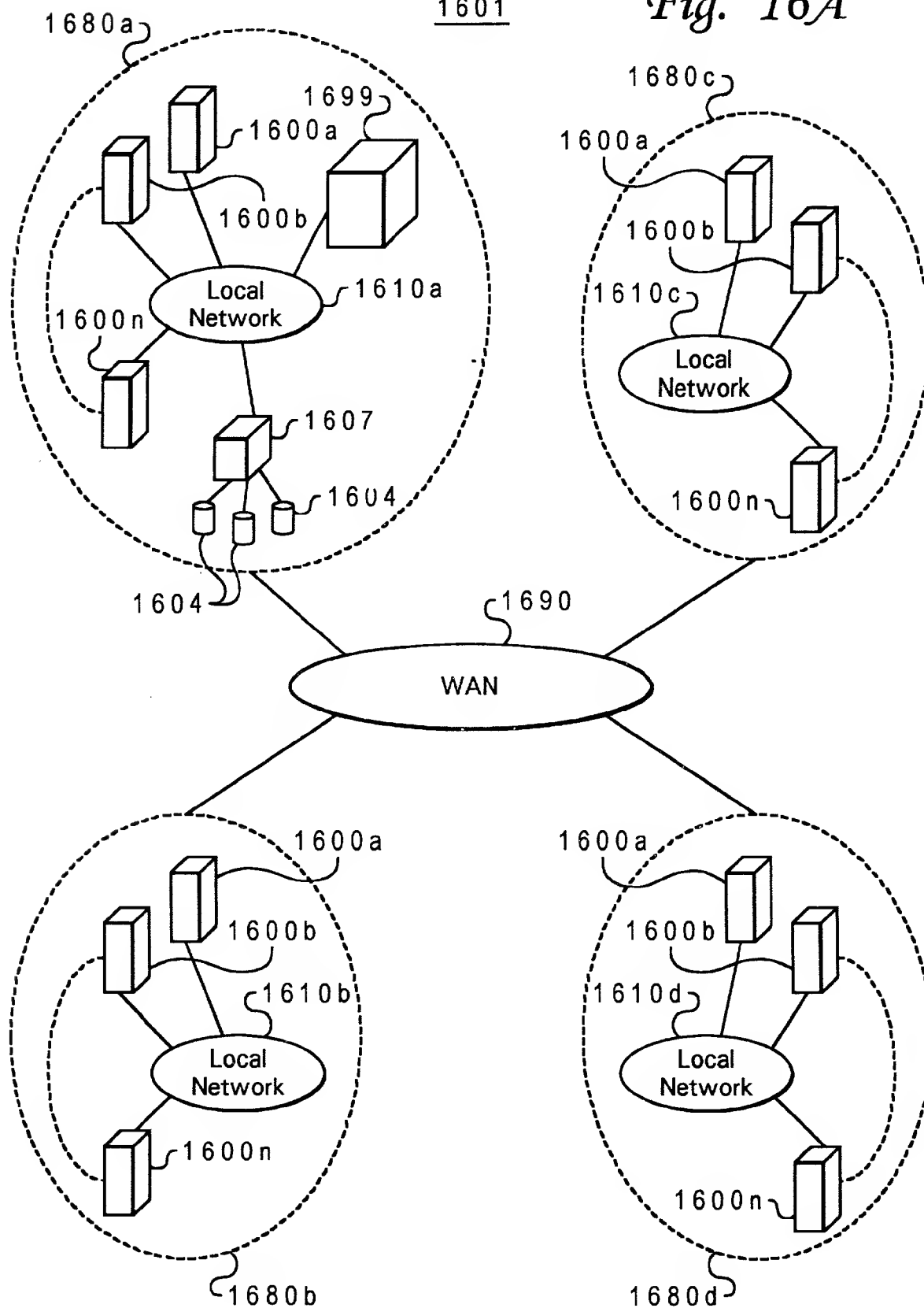


Fig. 15

35/62

1601

*Fig. 16A*



36/62

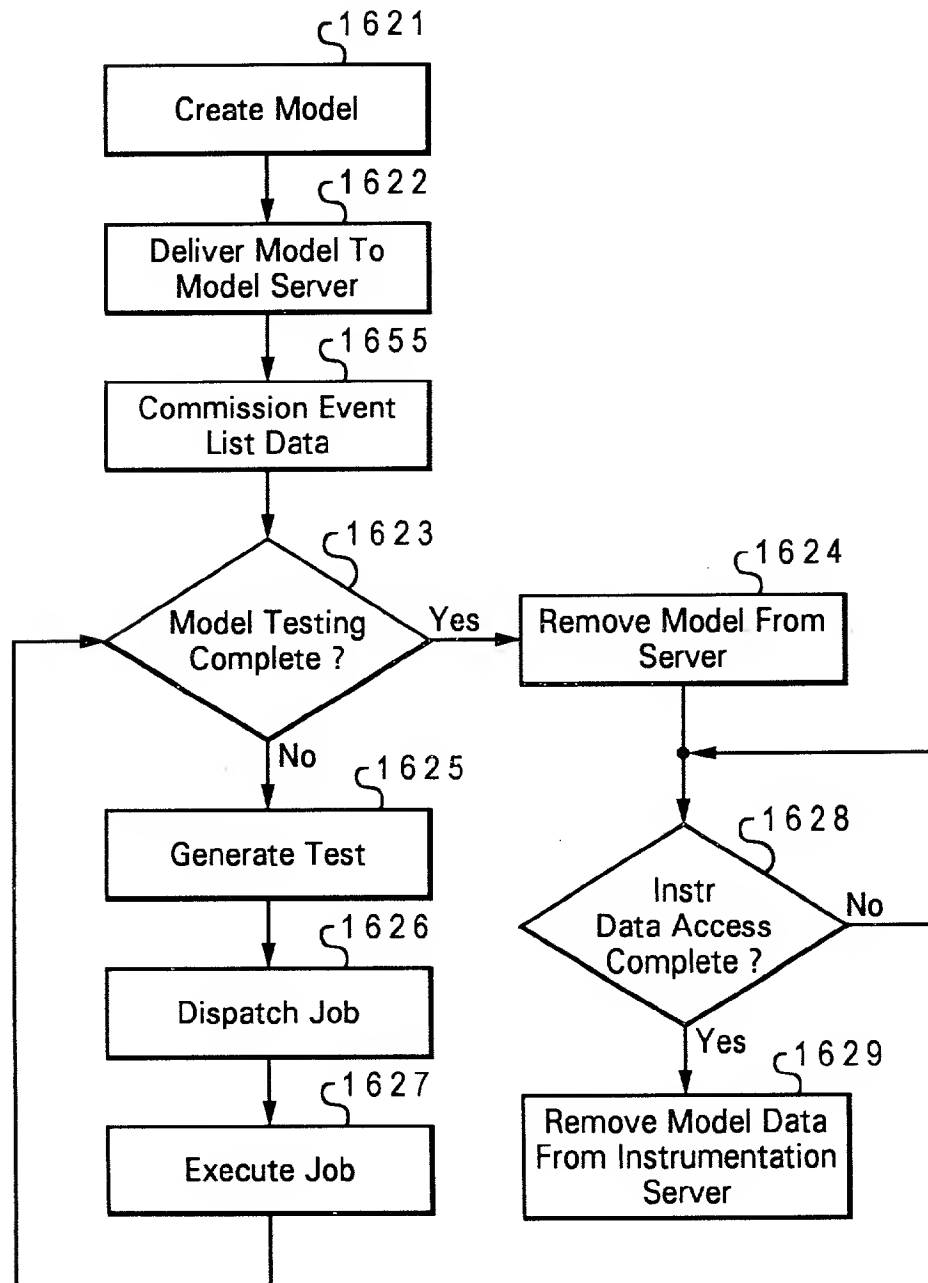


Fig. 16B

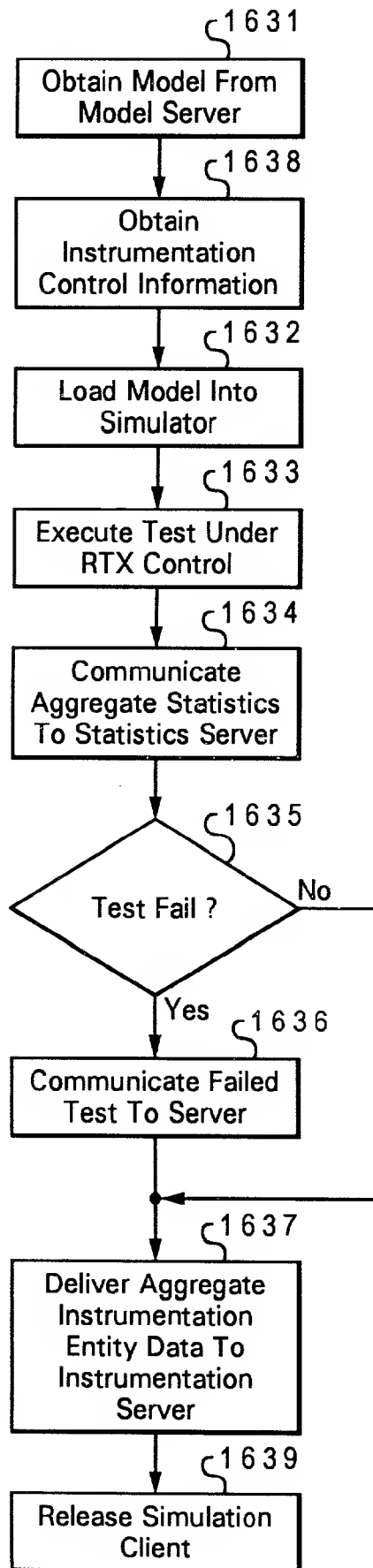


Fig. 16C

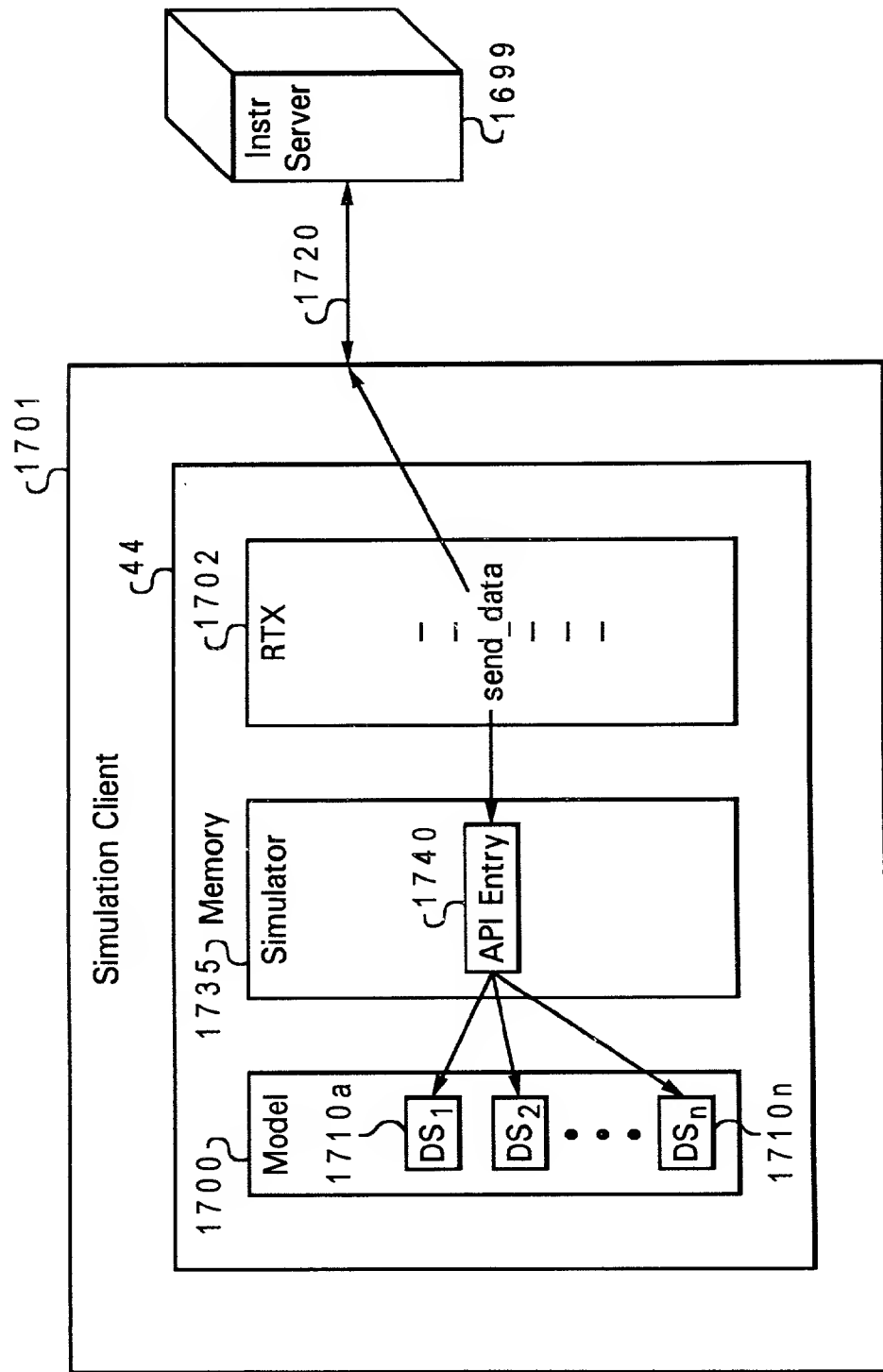
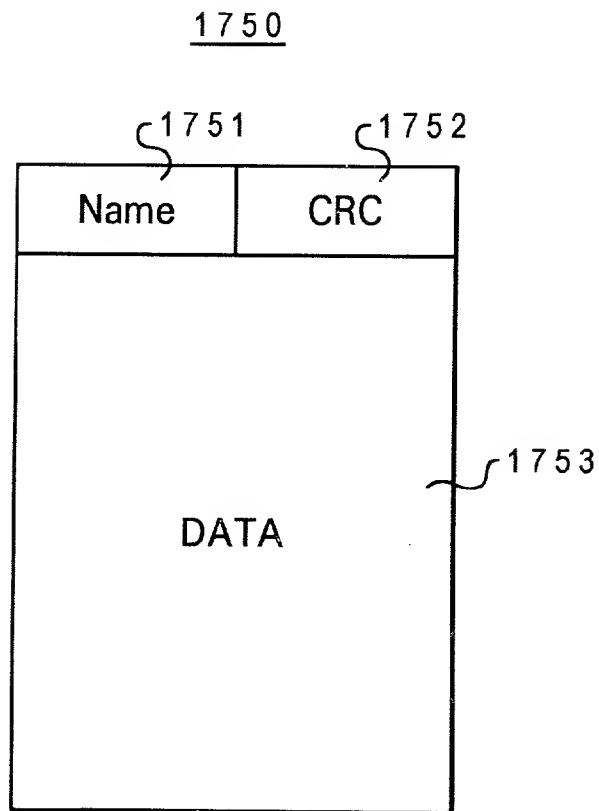


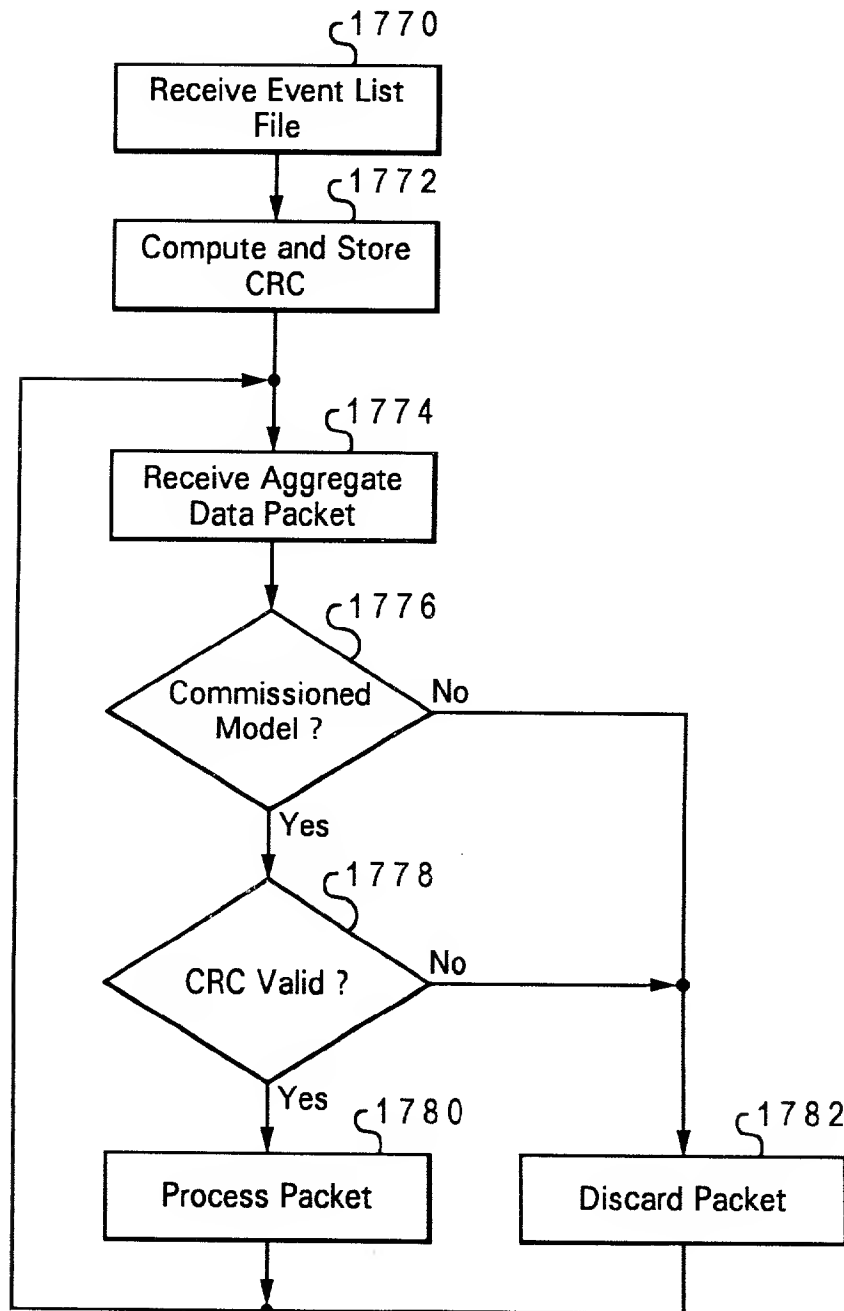
Fig. 17A

39/62



*Fig. 17B*

40/62

*Fig. 17C*



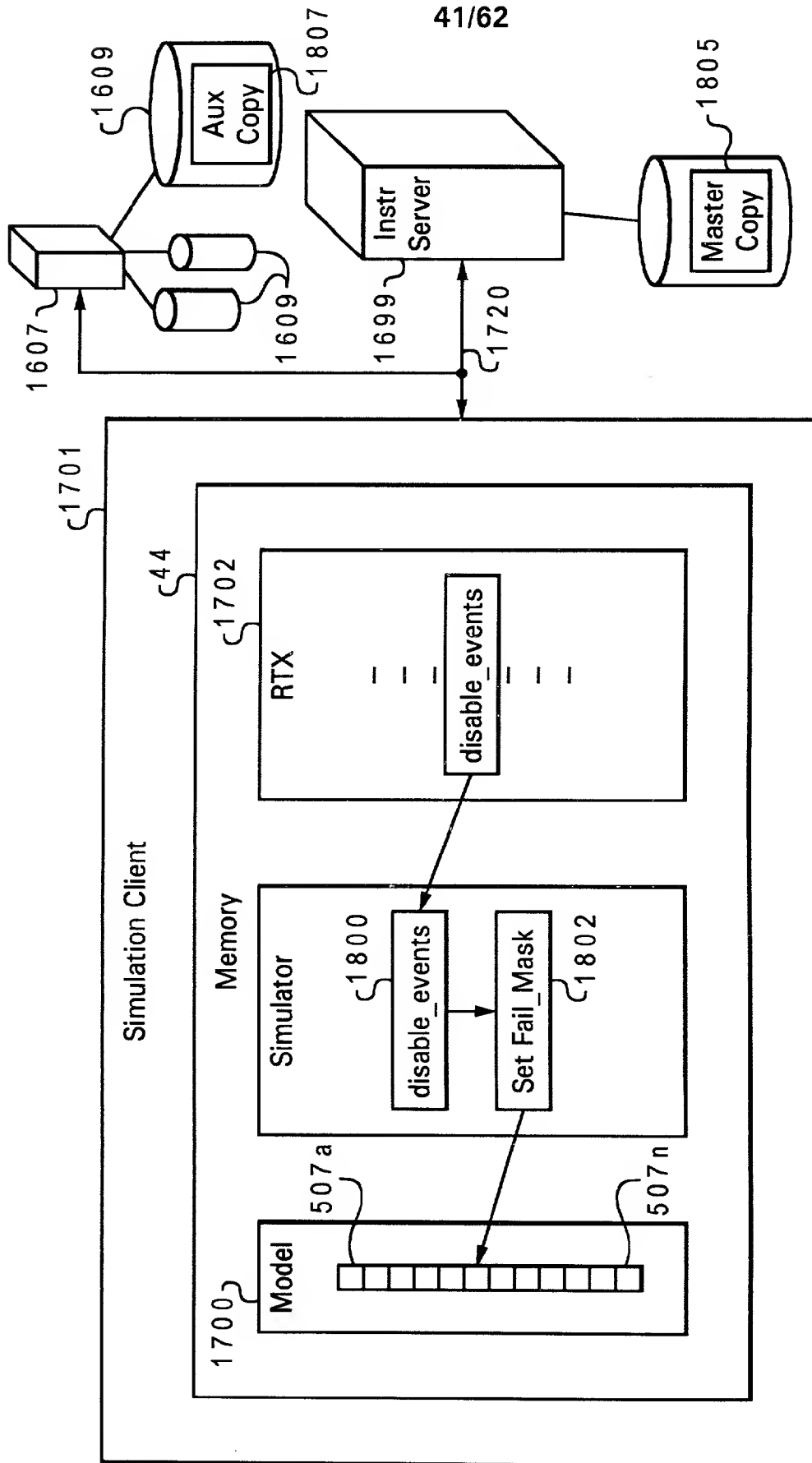
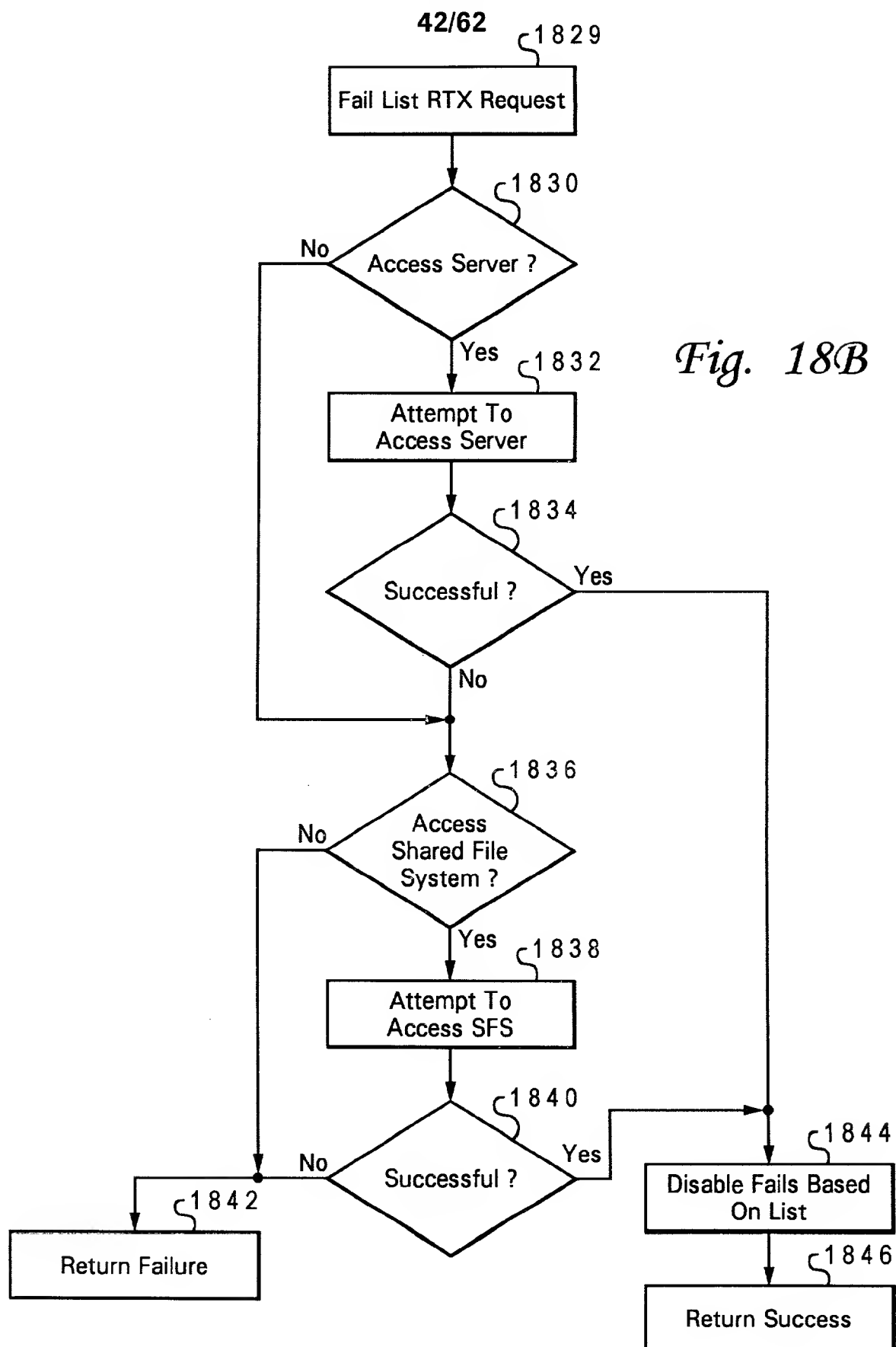


Fig. 18A



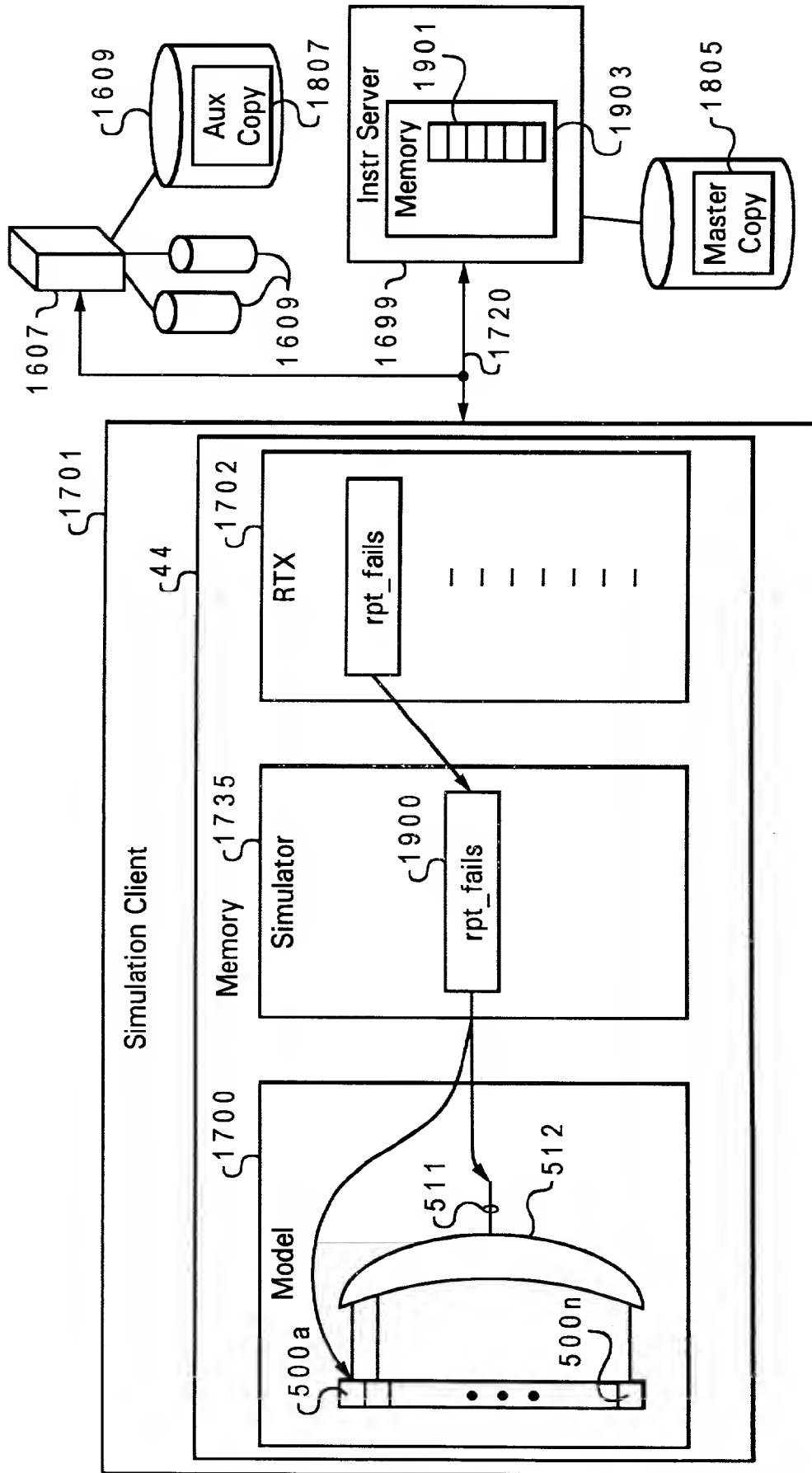
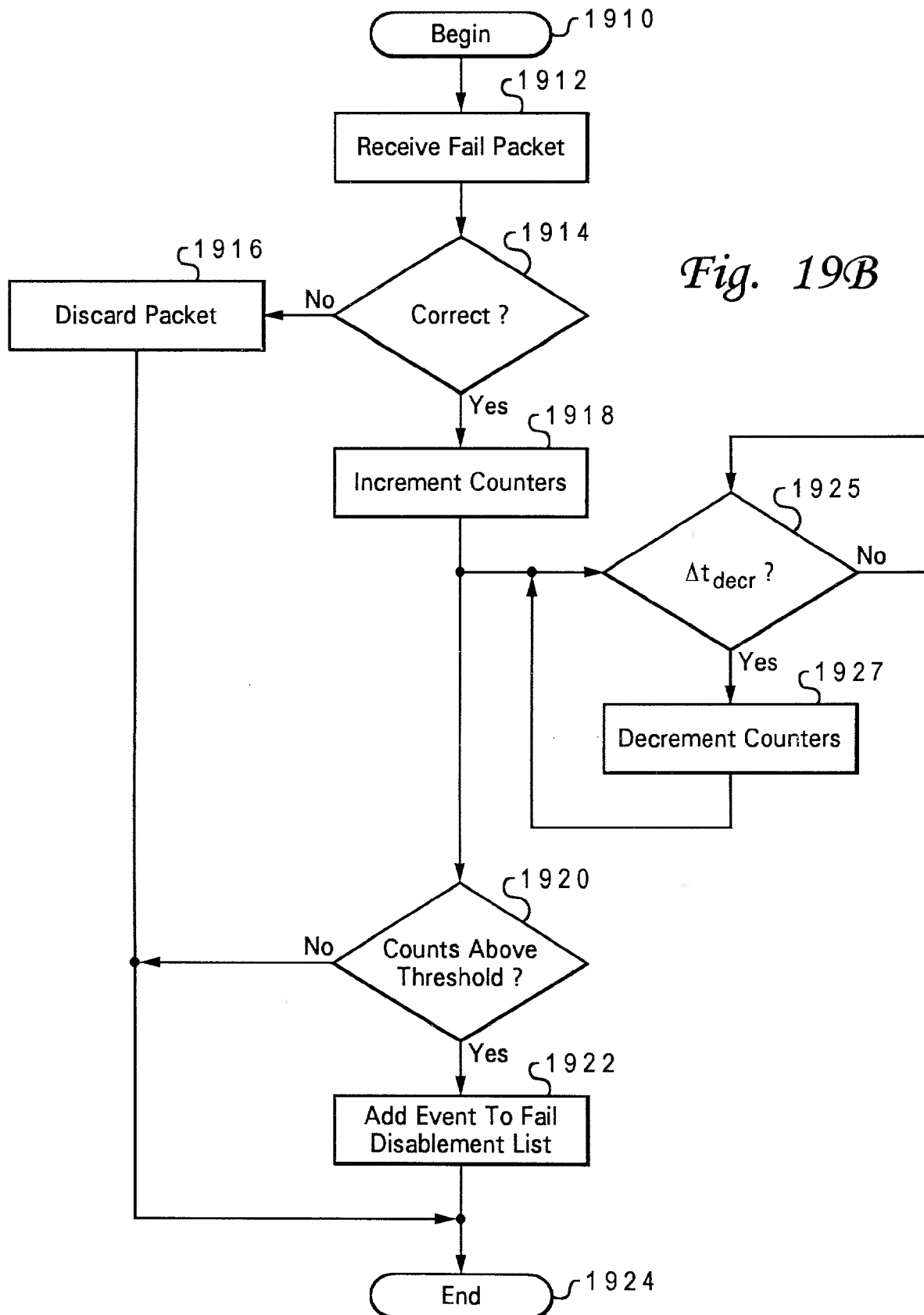


Fig. 19A

44/62



45/62

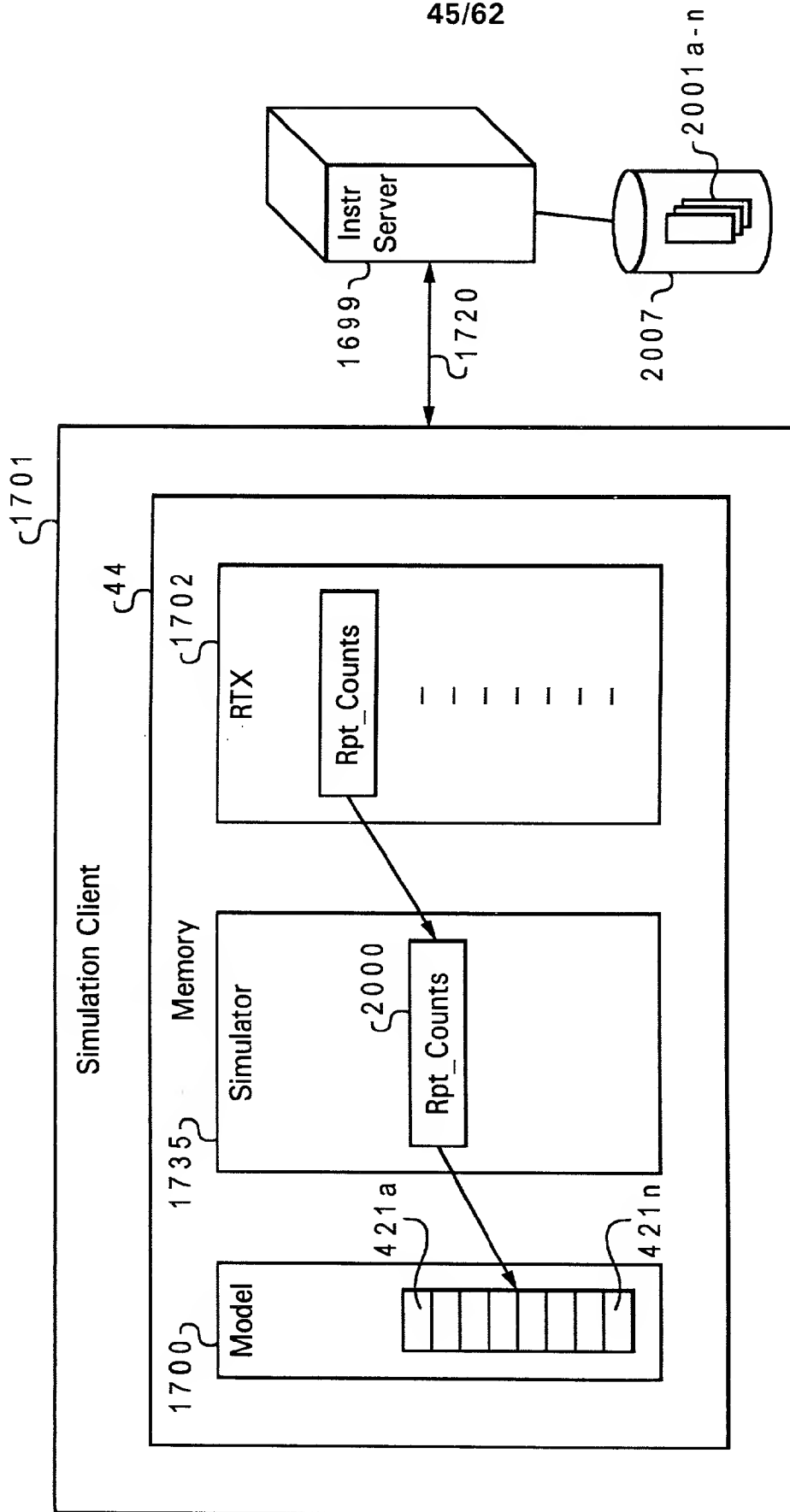


Fig. 20A

46/62

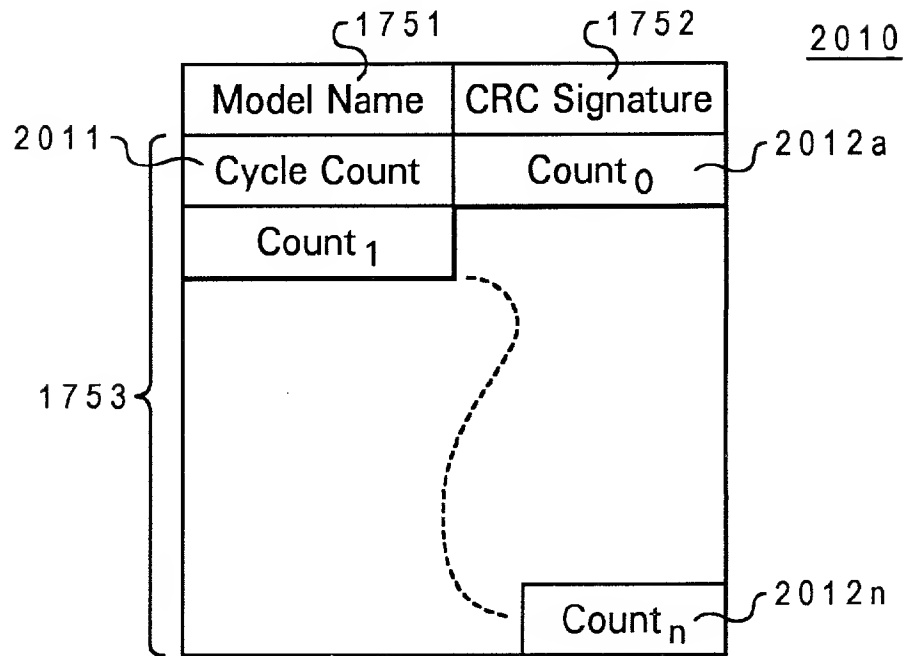


Fig. 20B

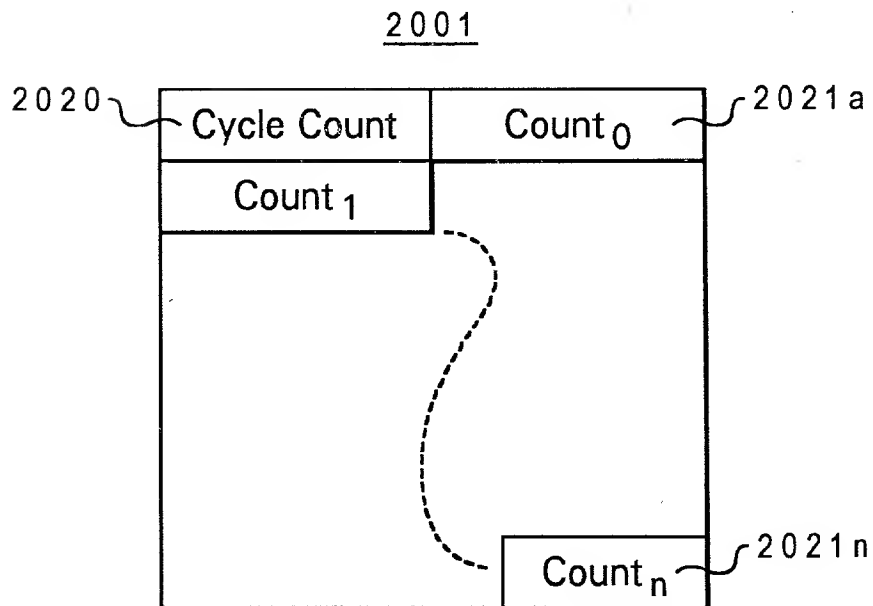


Fig. 20C

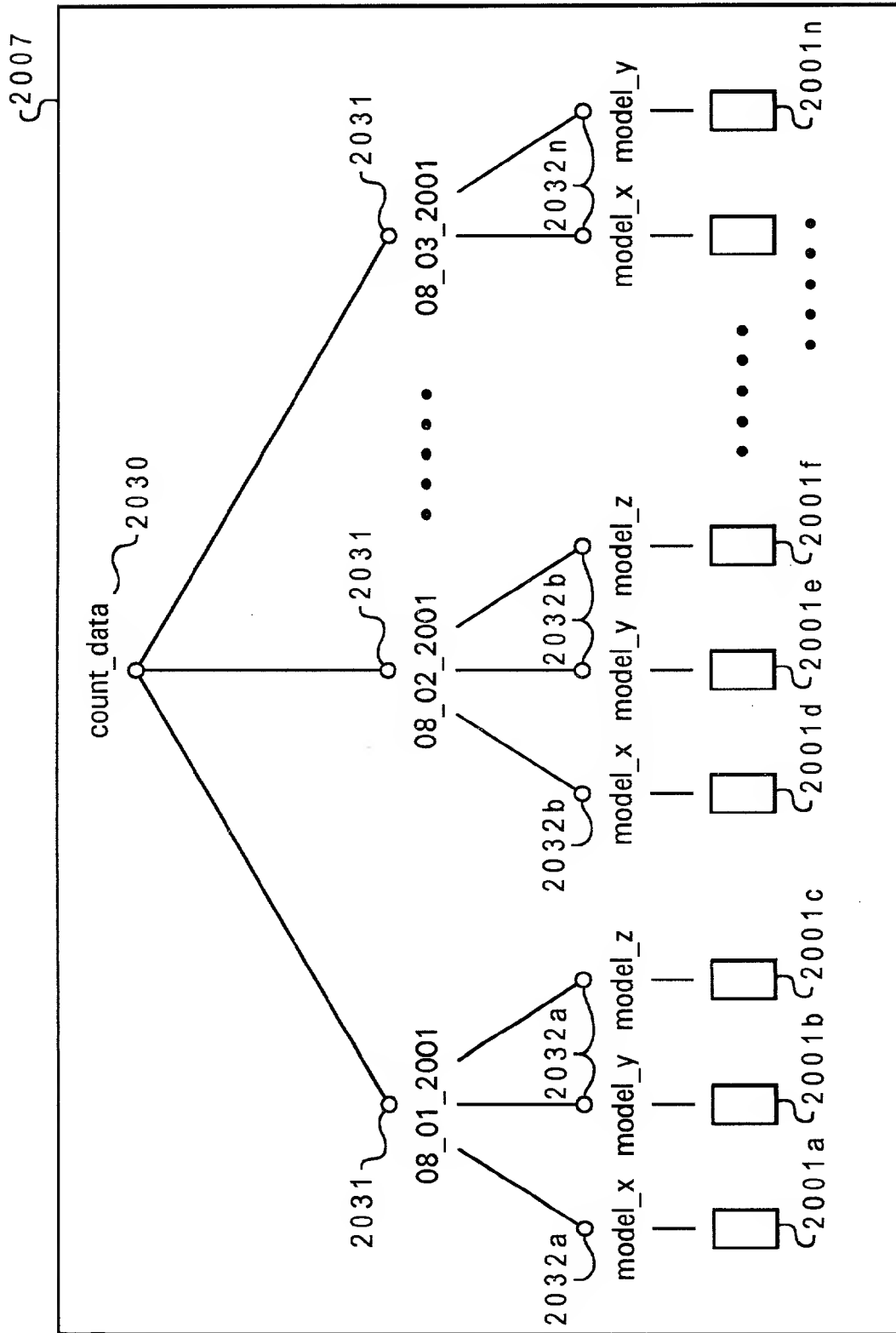
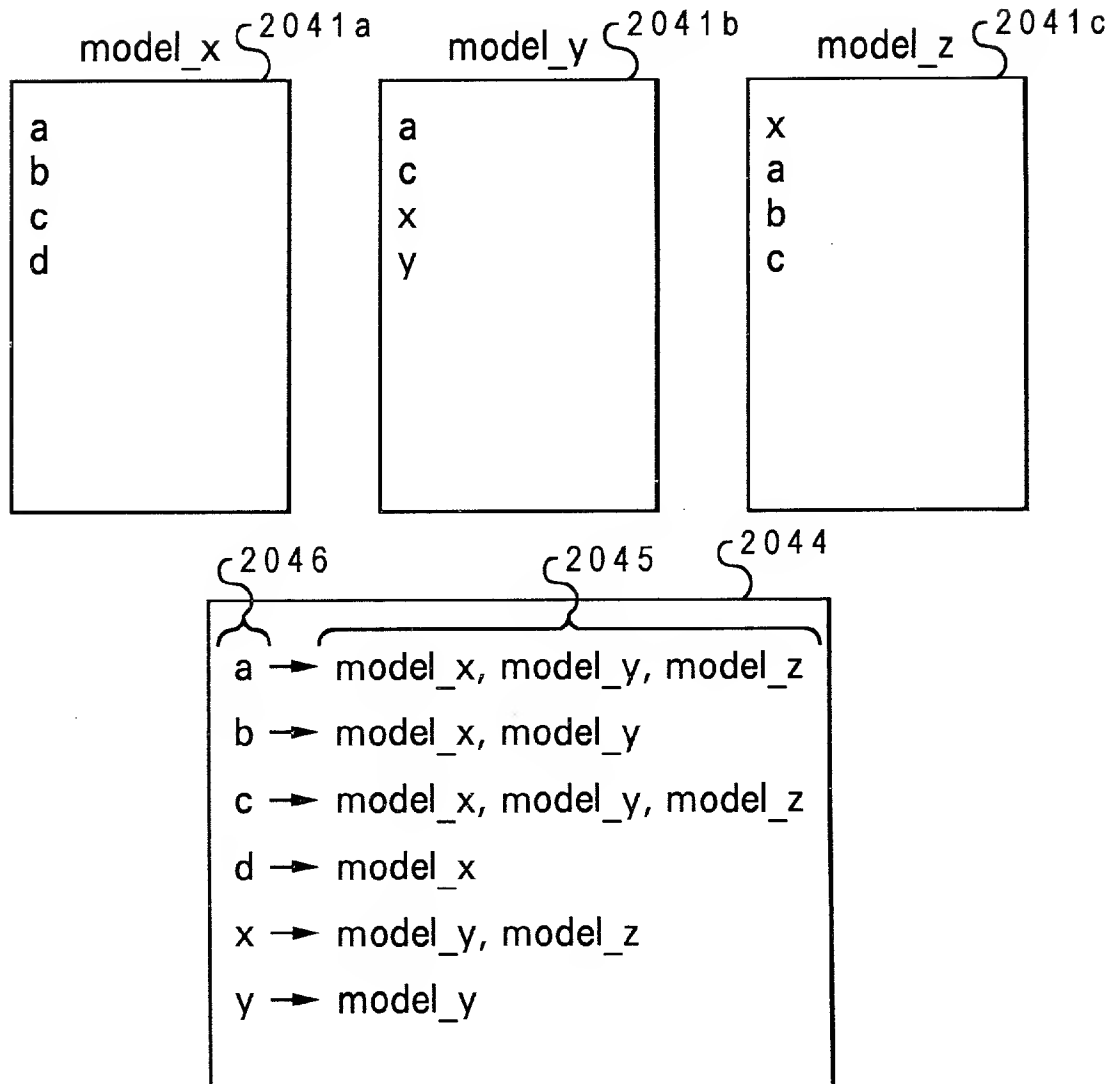


Fig. 20D

48/62



*Fig. 20E*



49/62

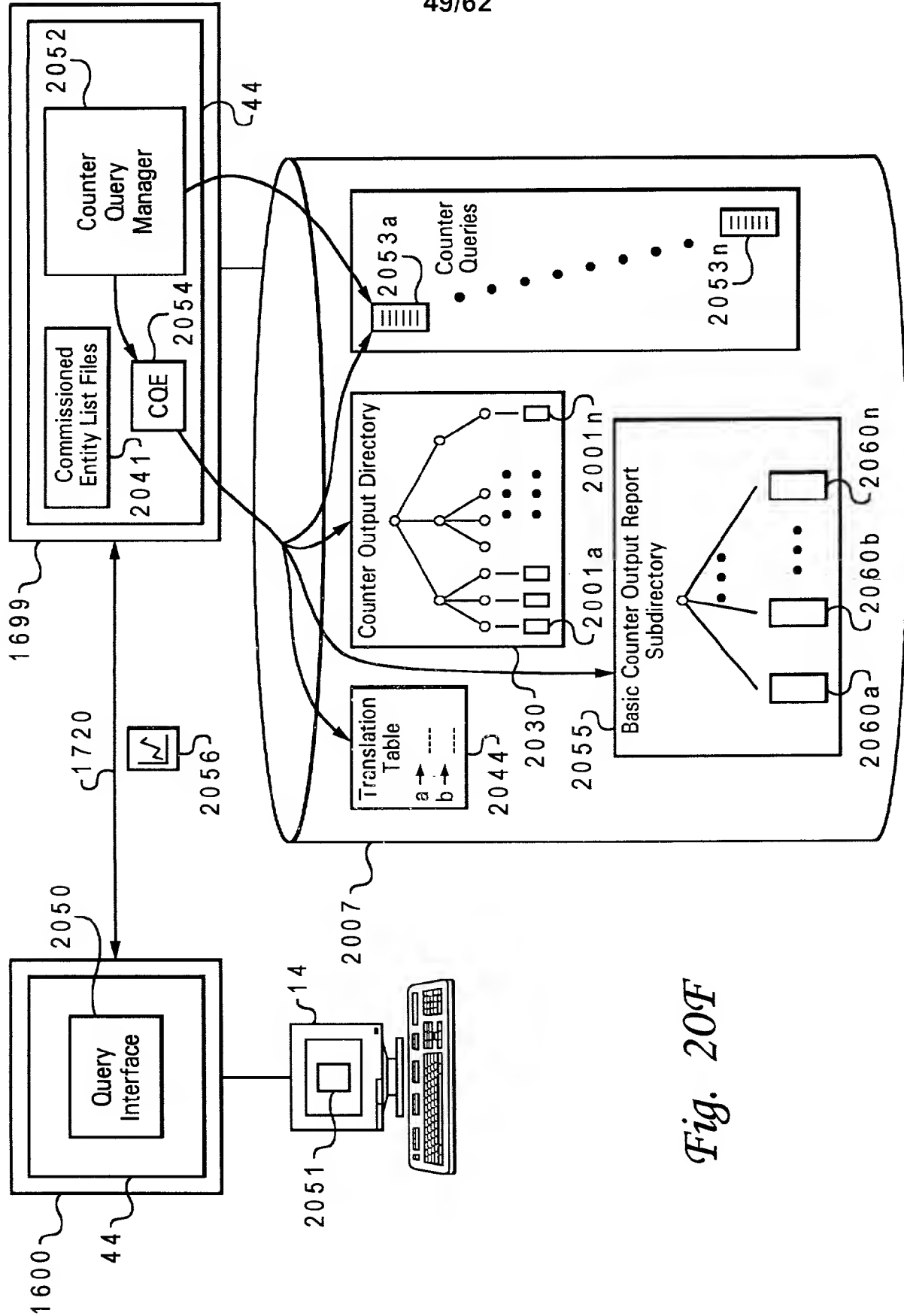
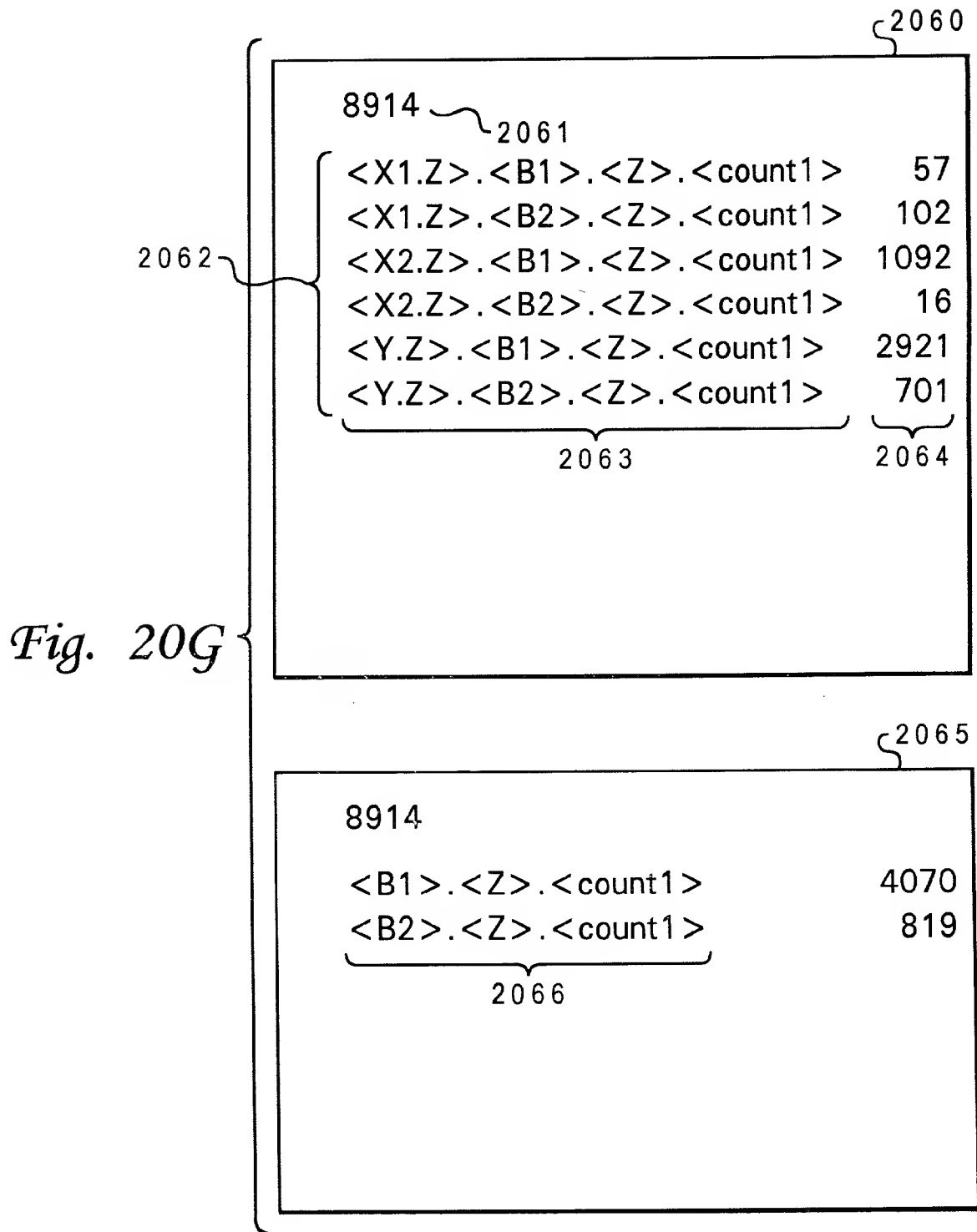
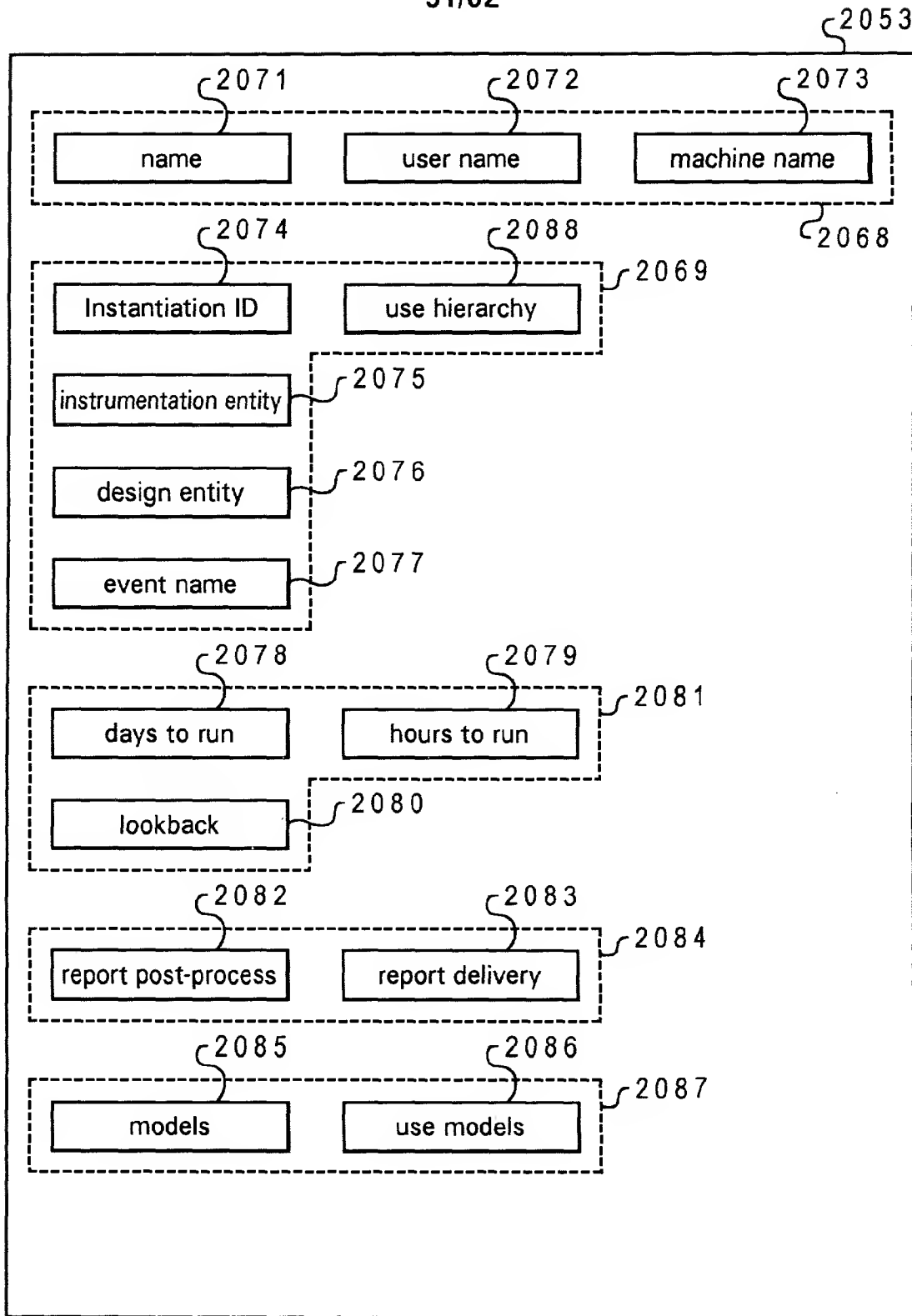


Fig. 20F

50/62



51/62

*Fig. 20H*

52/62

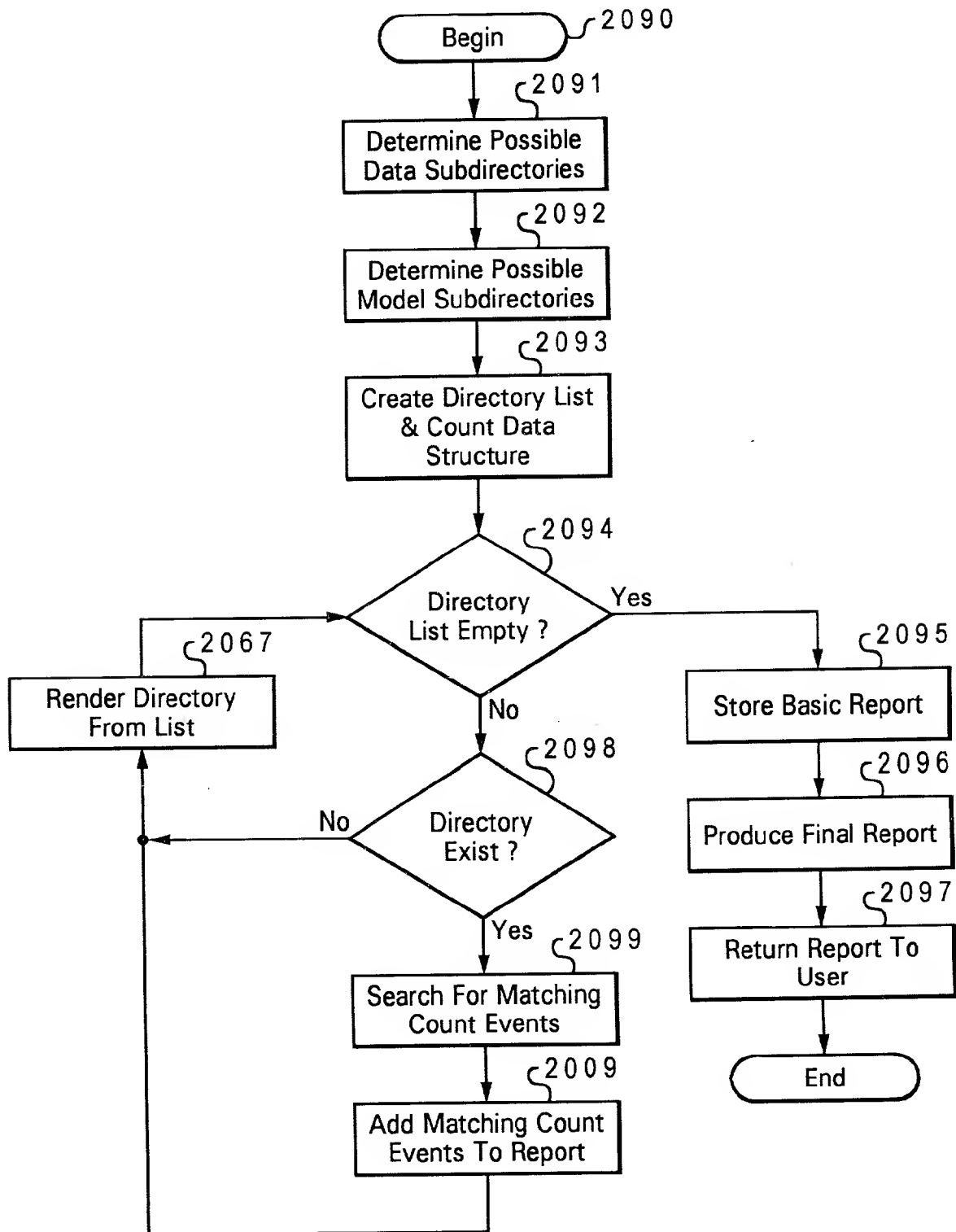


Fig. 20I

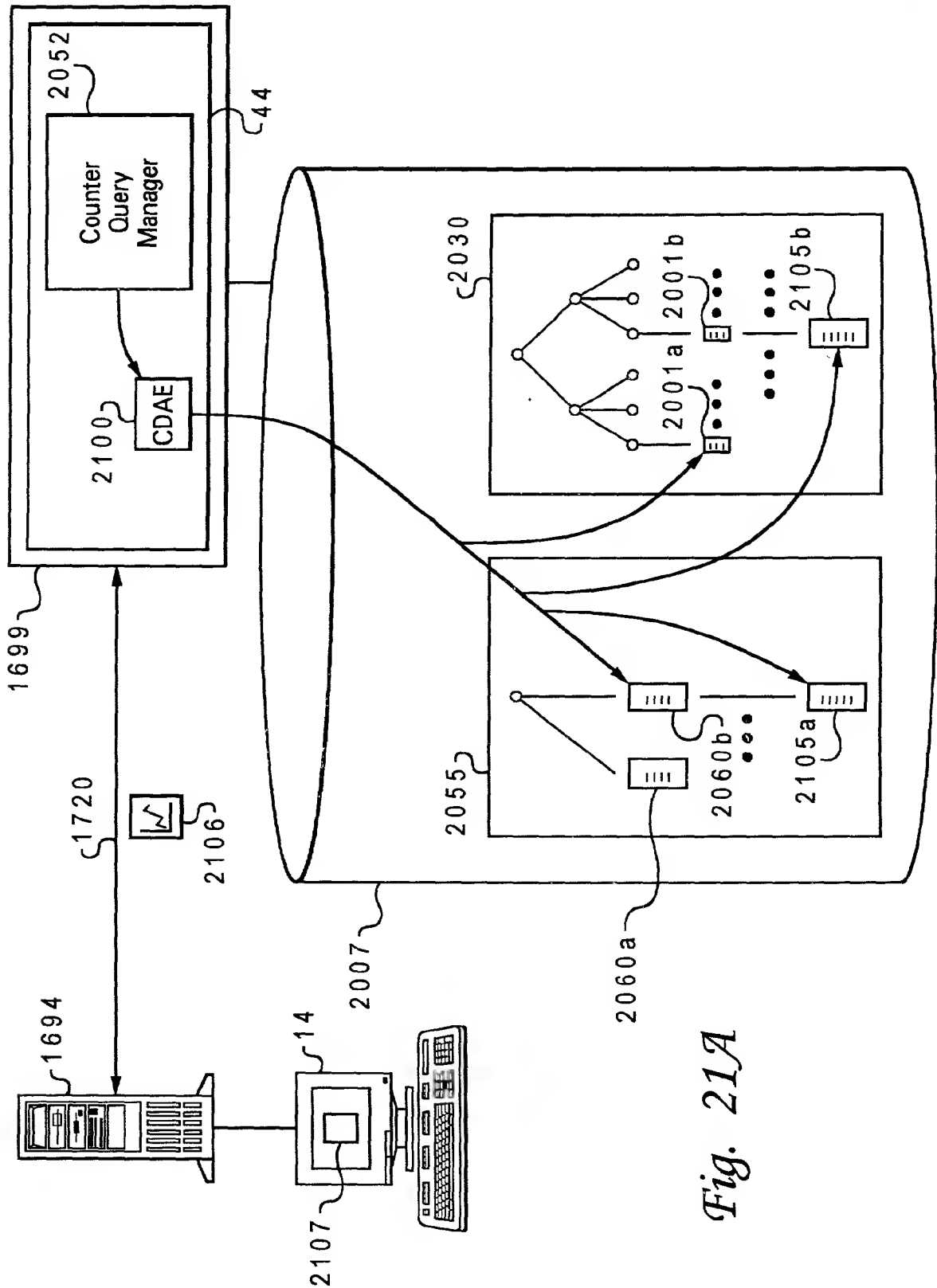


Fig. 21A

54/62

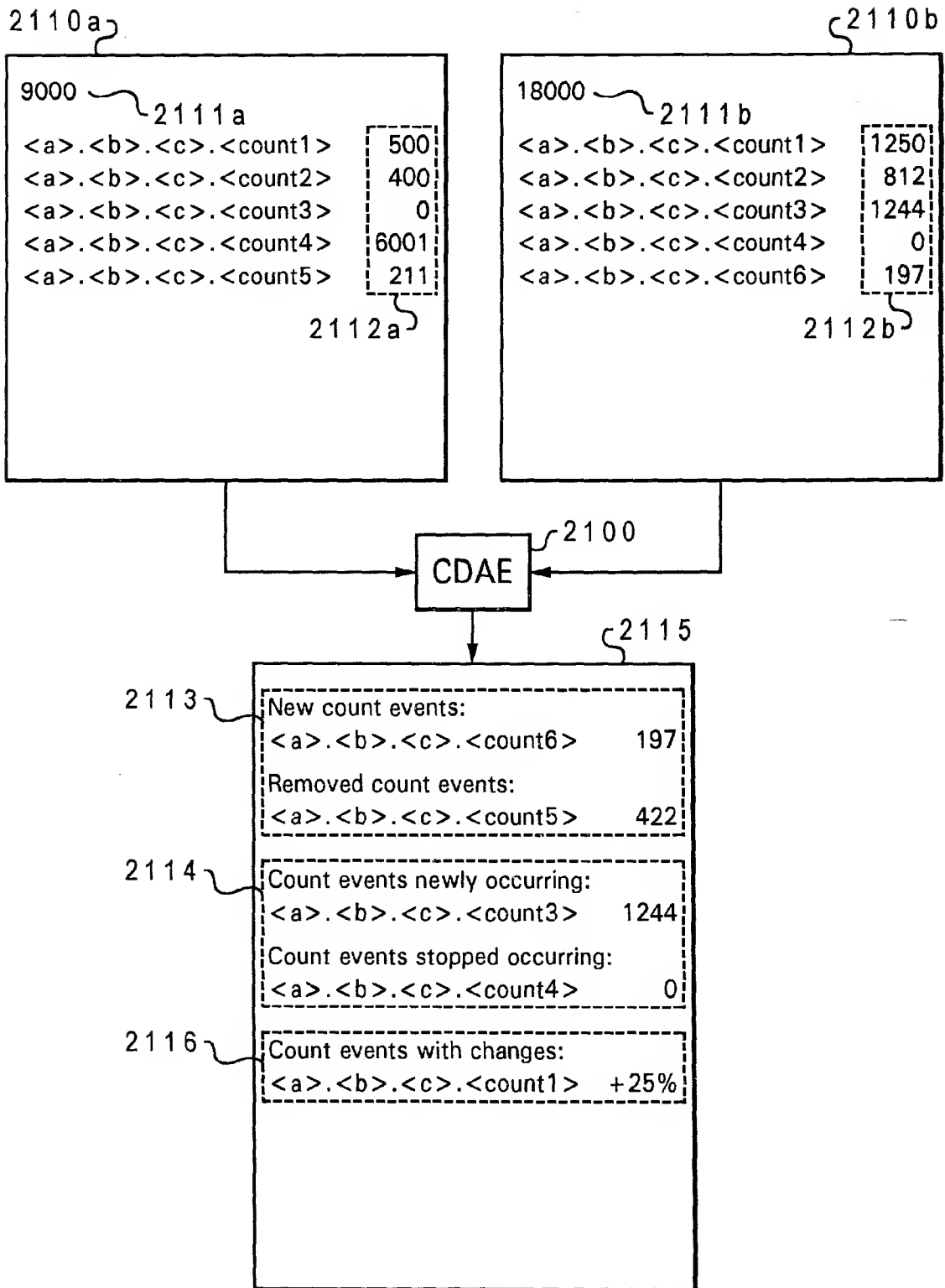


Fig. 21B

55/62

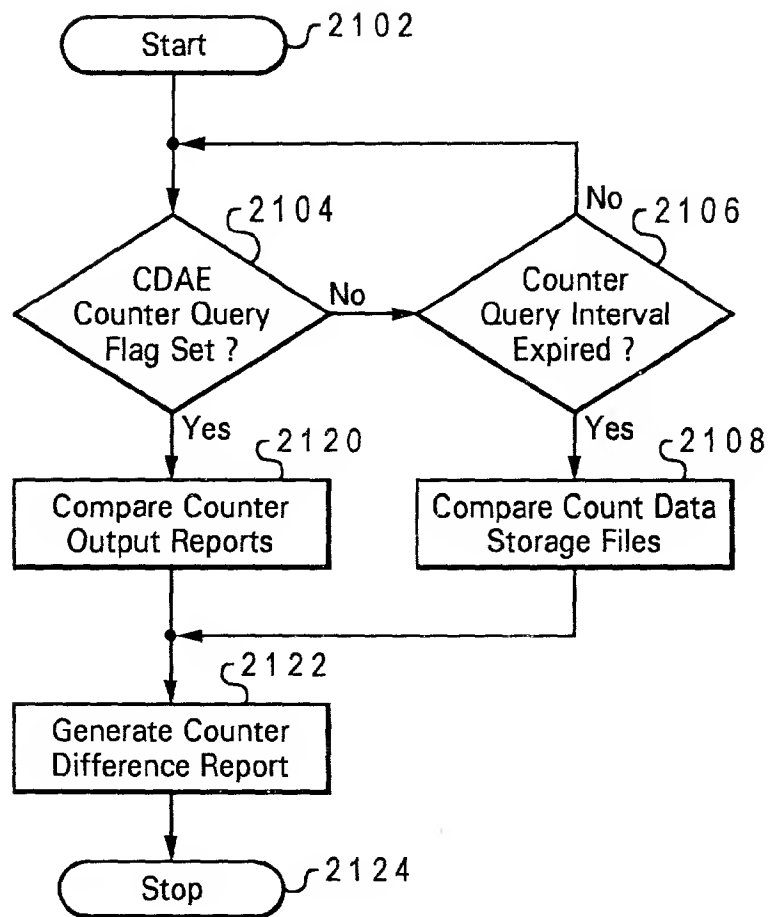
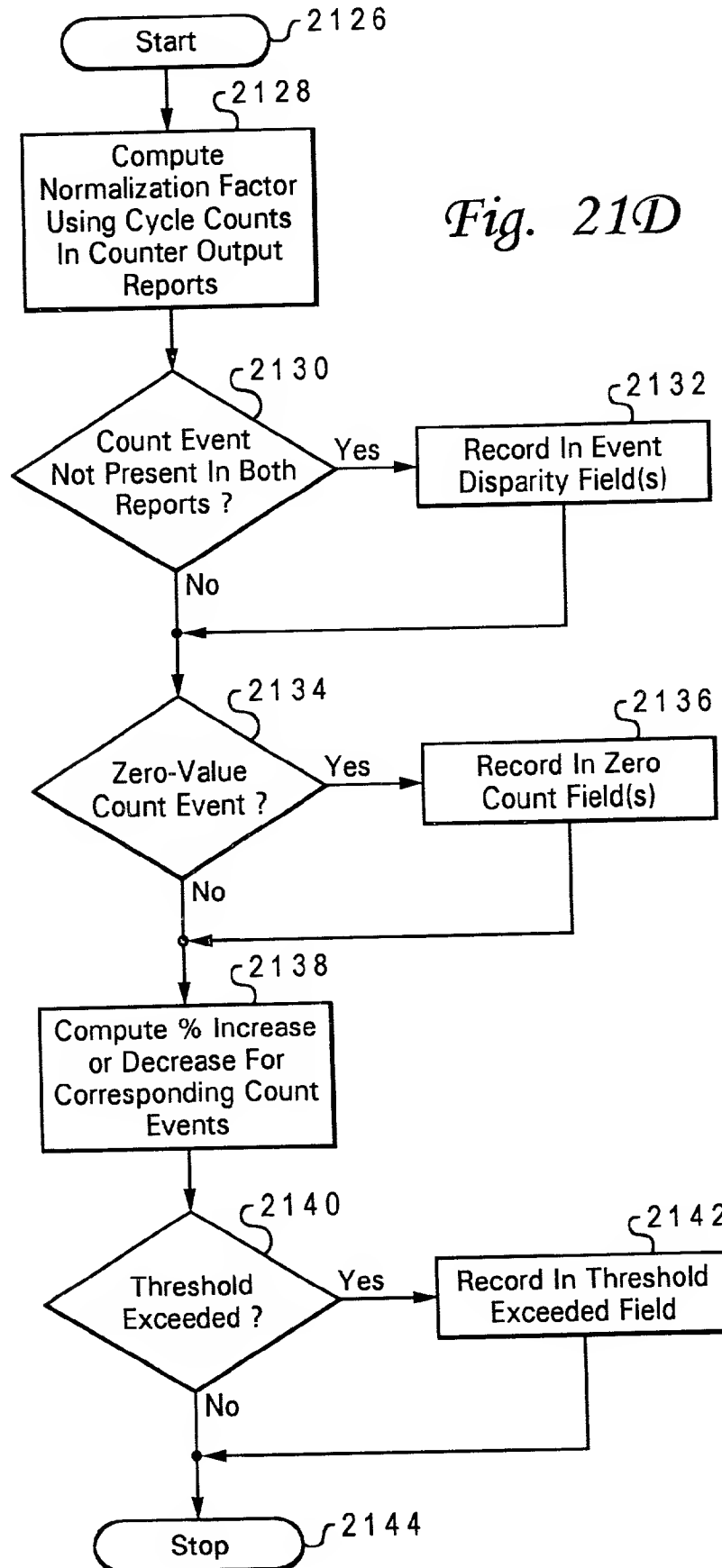


Fig. 21C

Fig. 21D





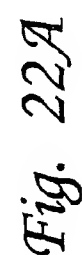


Fig. 22B

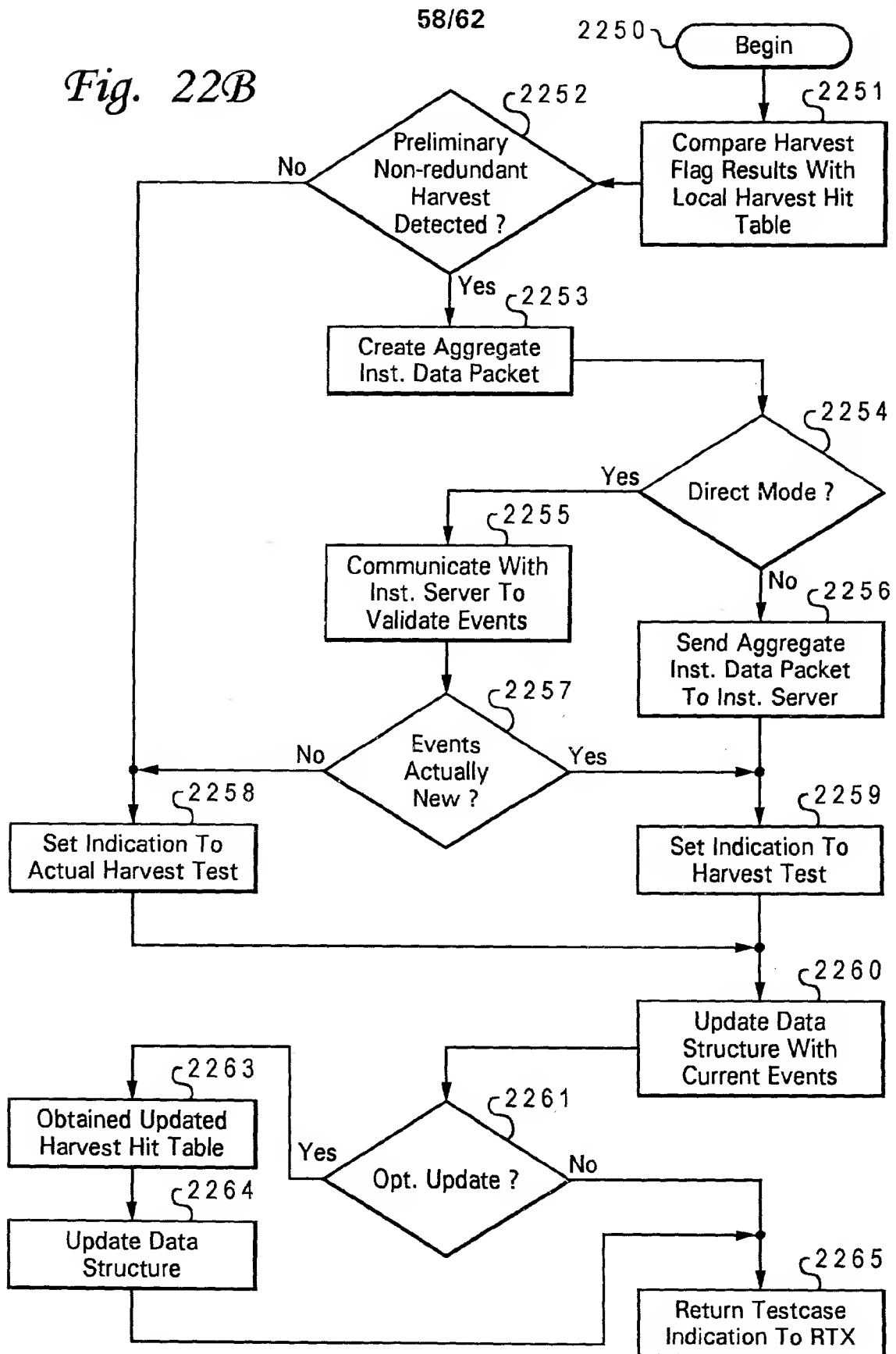
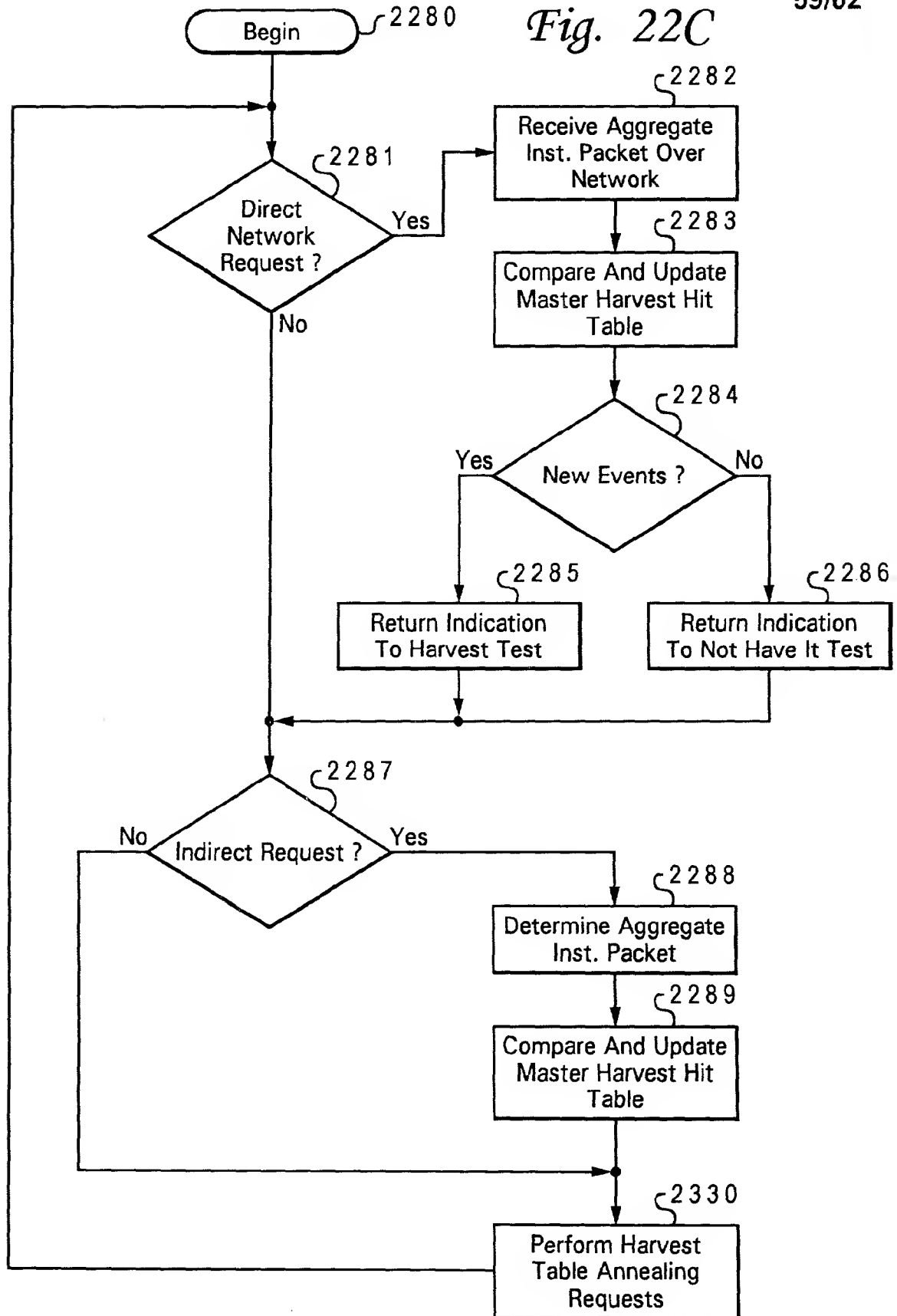


Fig. 22C



60/62

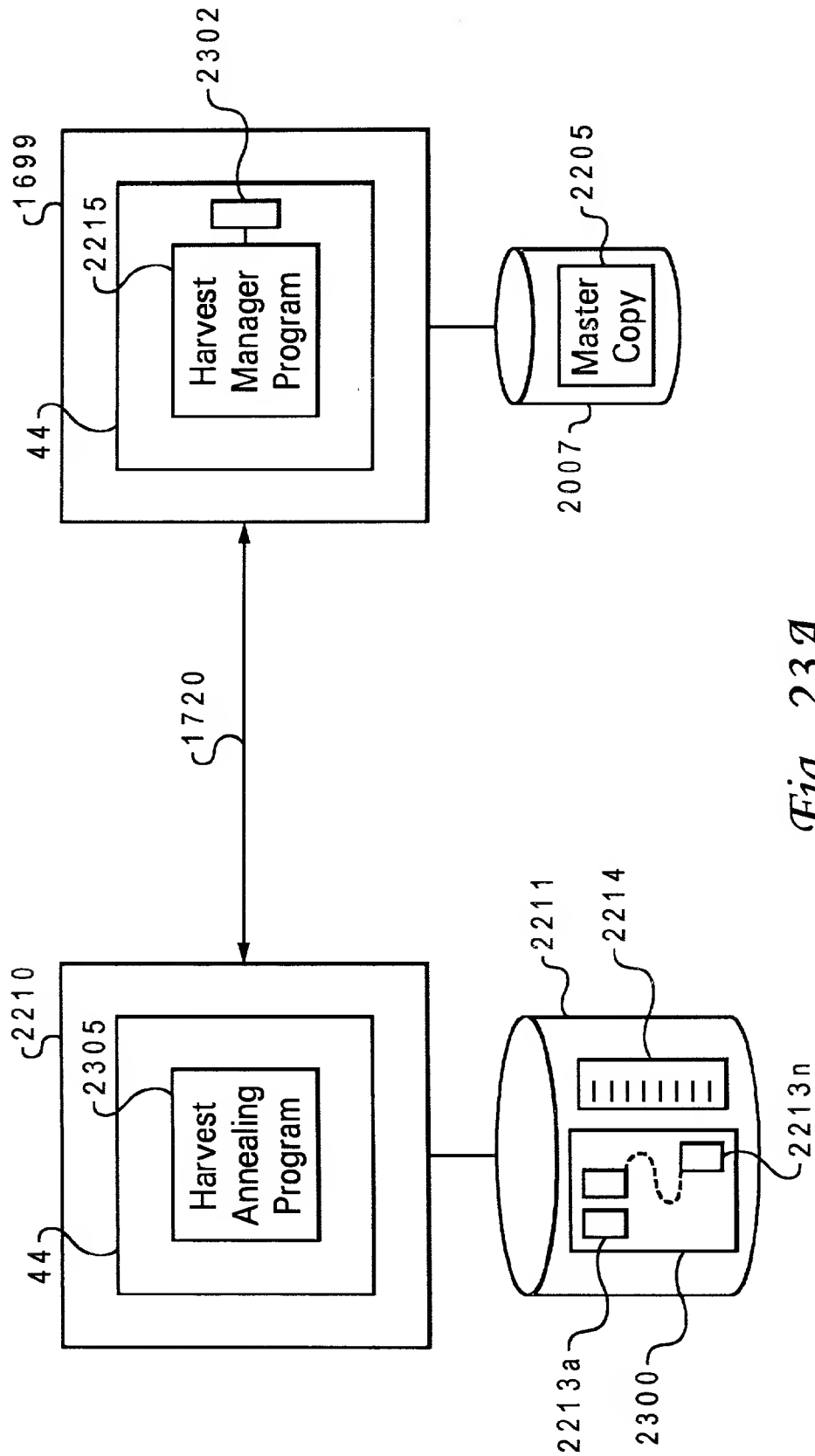
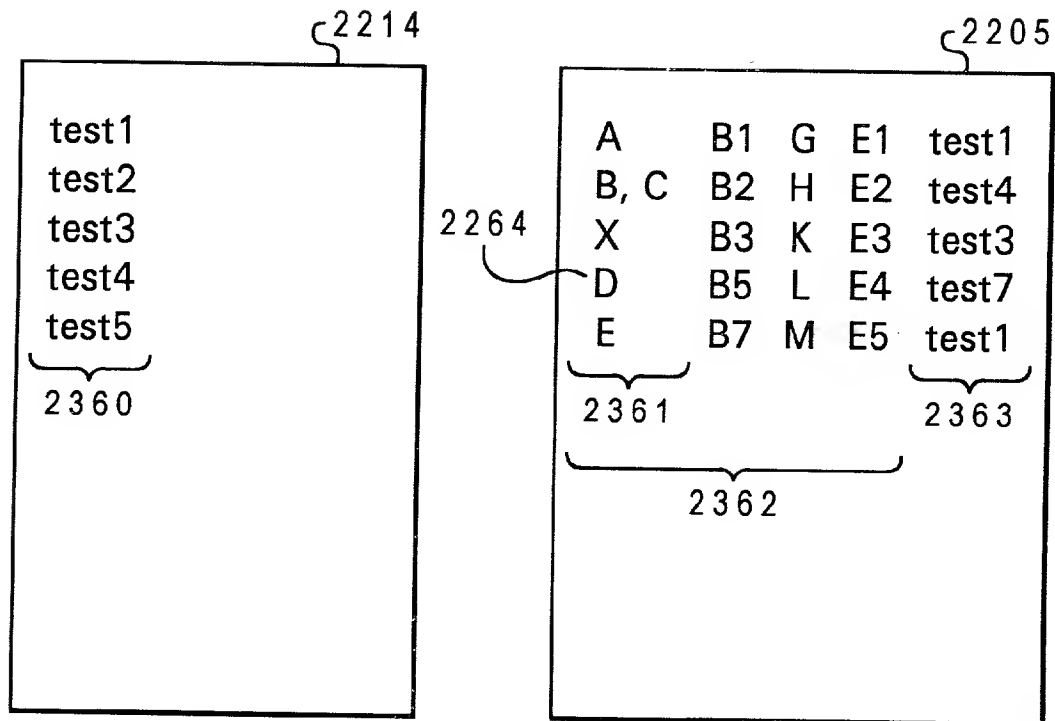


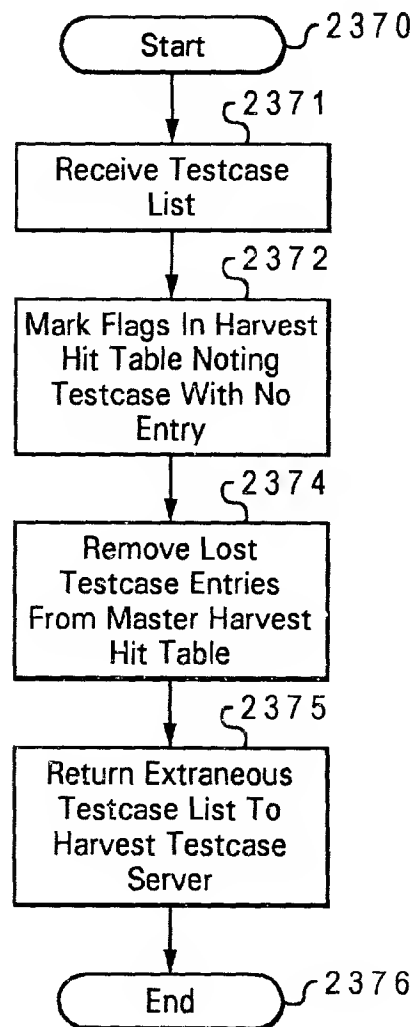
Fig. 23A

61/62



*Fig. 23B*

62/62

*Fig. 23C*